



(12) 发明专利

(10) 授权公告号 CN 108701118 B

(45) 授权公告日 2022.06.24

(21) 申请号 201780011171.6

(22) 申请日 2017.02.10

(65) 同一申请的已公布的文献号
申请公布号 CN 108701118 A

(43) 申请公布日 2018.10.23

(30) 优先权数据
62/293,922 2016.02.11 US

(85) PCT国际申请进入国家阶段日
2018.08.13

(86) PCT国际申请的申请数据
PCT/US2017/017371 2017.02.10

(87) PCT国际申请的公布数据
W02017/139575 EN 2017.08.17

(73) 专利权人 电子湾有限公司
地址 美国加利福尼亚州

(72) 发明人 刘明宽

(74) 专利代理机构 中科专利商标代理有限责任
公司 11021

专利代理师 倪斌

(51) Int.Cl.
G06F 40/30 (2020.01)
G06N 5/02 (2006.01)
G06N 20/00 (2019.01)
G06F 12/08 (2016.01)

(56) 对比文件
US 8473532 B1, 2013.06.25
US 8473532 B1, 2013.06.25
US 9224386 B1, 2015.12.29
JP 2015005027 A, 2015.01.08
CN 101281520 A, 2008.10.08
CN 101251841 A, 2008.08.27
CN 102156686 A, 2011.08.17
CN 102439590 A, 2012.05.02

审查员 靳超

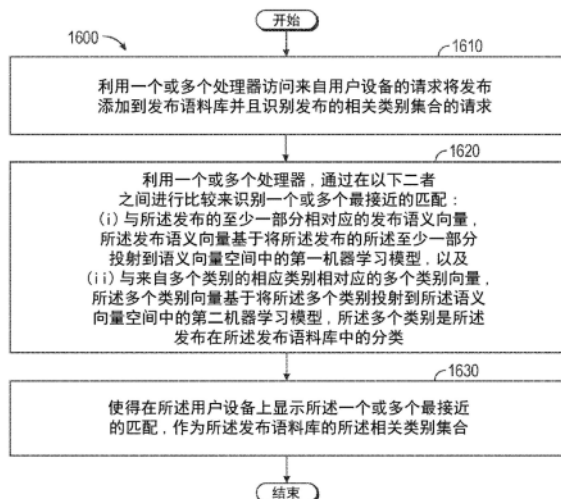
权利要求书2页 说明书26页 附图21页

(54) 发明名称

语义类别分类

(57) 摘要

根据示例实施例,描述了基于序列语义嵌入和并行学习的大规模类别分类。在一个示例中,通过在以下二者之间进行比较来识别一个或多个最接近的匹配:(i)与发布的至少一部分相对应的发布语义向量,所述发布语义向量基于将所述发布的所述至少一部分投射到语义向量空间中的第一机器学习模型,以及(ii)与来自多个类别的相应类别相对应的多个类别向量。



1. 一种用于语义类别分类的方法,包括:

利用一个或多个处理器访问来自用户设备的请求将发布添加到发布语料库并且识别所述发布的相关类别集合的请求;

利用所述一个或多个处理器,通过在以下二者之间进行比较来识别一个或多个最接近的匹配:(i) 嵌入所述发布的至少一部分的语义含义的发布语义向量,所述发布语义向量基于嵌入所述发布的所述至少一部分的语义含义并投射到语义向量空间中的第一机器学习模型,以及(ii) 嵌入来自多个类别的相应类别的语义含义的多个类别向量,所述多个类别向量基于嵌入所述多个类别的语义含义并投射到所述语义向量空间中的第二机器学习模型,所述多个类别是所述发布在所述发布语料库中的分类;以及

使得在所述用户设备上显示所述一个或多个最接近的匹配,作为所述发布语料库的所述相关类别集合。

2. 根据权利要求1所述的方法,其中,所述类别是叶类别。

3. 根据权利要求1所述的方法,其中,所述类别是在所述多个类别的类别树中的根级之下的至少两个树级的类别路径。

4. 根据权利要求1所述的方法,其中,所述发布的所述至少一部分包括所述发布的标题。

5. 根据权利要求1所述的方法,其中,在从所述发布语料库的先前添加的发布中自动导出的数据上训练所述第一机器学习模型和所述第二机器学习模型中的至少一个。

6. 根据权利要求1所述的方法,其中,以子词级和字符级别中的一个或多个训练所述第一机器学习模型和所述第二机器学习模型中的至少一个,以减少运行时的词汇外术语。

7. 根据权利要求1所述的方法,还包括:

向所述多个类别添加新类别,而不在所述新类别上重新训练所述第二机器学习模型,其中,被识别为一个或多个最接近的匹配的所述一个或多个最接近的匹配包括所述新类别。

8. 一种用于语义类别分类的计算机,包括:

存储指令的存储设备;以及

一个或多个硬件处理器,由所述指令配置为执行包括以下各项的操作:

利用一个或多个处理器访问来自用户设备的请求将发布添加到发布语料库并且识别所述发布的相关类别集合的请求;

利用所述一个或多个处理器,通过在以下二者之间进行比较来识别一个或多个最接近的匹配:(i) 嵌入所述发布的至少一部分的语义含义的发布语义向量,所述发布语义向量基于嵌入所述发布的所述至少一部分的语义含义并投射到语义向量空间中的第一机器学习模型,以及(ii) 嵌入来自多个类别的相应类别的语义含义的多个类别向量,所述多个类别向量基于嵌入所述多个类别的语义含义并投射到所述语义向量空间中的第二机器学习模型,所述多个类别是所述发布在所述发布语料库中的分类;以及

使得在所述用户设备上显示所述一个或多个最接近的匹配,作为所述发布语料库的所述相关类别集合。

9. 根据权利要求8所述的计算机,其中,所述类别是叶类别。

10. 根据权利要求8所述的计算机,其中,所述类别是在所述多个类别的类别树中的根

级之下的至少两个树级的类别路径。

11. 根据权利要求8所述的计算机, 其中, 所述发布的所述至少一部分包括所述发布的标题。

12. 根据权利要求8所述的计算机, 其中, 在从所述发布语料库的先前添加的发布中自动导出的数据上训练所述第一机器学习模型和所述第二机器学习模型中的至少一个。

13. 根据权利要求8所述的计算机, 其中, 以子词级和字符级别中的一个或多个训练所述第一机器学习模型和所述第二机器学习模型中的至少一个, 以减少运行时的词汇外术语。

14. 根据权利要求8所述的计算机, 所述操作还包括:

向所述多个类别添加新类别, 而不在所述新类别上重新训练所述第二机器学习模型, 其中, 被识别为一个或多个最接近的匹配的所述一个或多个最接近的匹配包括所述新类别。

15. 一种携带机器可读指令的机器可读介质, 所述机器可读指令在被机器的一个或多个处理器执行时, 使所述机器执行根据权利要求1至7中任一项所述的方法。

语义类别分类

[0001] 相关申请的交叉引用

[0002] 本申请要求2016年2月11日提交的美国临时申请No.62/293,922的优先权,其全部内容通过引用并入本文。

技术领域

[0003] 本公开的实施例总体上涉及基于序列语义嵌入和并行学习的大规模类别分类和推荐系统(CatReco)。

背景技术

[0004] 发布语料库中的发布的适当分类对于帮助系统响应于用户的查询来提供发布(例如,产品和/或服务)推荐是重要的。系统使用发布描述来对发布编索引,使得潜在用户可以通过用户的查询来定位发布。

附图说明

[0005] 各个所附附图仅示出了本公开示例实施例,并且不可以被认为限制其范围。

[0006] 图1是示出了根据一些示例实施例的联网系统的框图。

[0007] 图2是根据示例实施例,更详细地示出图1的列表系统的框图。

[0008] 图3A和图3B是根据示例实施例的来自列表系统的用户界面,该用户界面用于提供列表标题并且为列表标题选择类别。

[0009] 图4示出了将源语义向量与最接近的目标语义向量进行匹配的简单示例。

[0010] 图5A示出了使用SSE(序列语义嵌入)为用户至少提供一个CatReco 的流程图。

[0011] 图5B示出了根据示例实施例的使用SSE向服务提供叶类别(LeafCat)标识(ID)的回调集流程图。

[0012] 图6A示出了根据示例实施例的用于执行运行时过程的流程图,该运行时过程用于执行针对基本SSE CatReco服务的运行时分类过程。

[0013] 图6B示出了根据示例实施例的用于执行离线过程的流程图,该离线过程用于预先计算基本SSE CatReco服务的目标的语义向量。

[0014] 图6C示出了根据另一示例实施例的用于执行基本SSE CatReco服务的运行时分类过程的流程图。

[0015] 图6D示出了根据示例实施例的用于执行基本SSE CatReco服务(包括在线和离线组件)的流程图。

[0016] 图7示出了根据示例实施例的用来训练用于基本SSE CatReco服务的SSE模型的方法的流程图。

[0017] 图8示出了根据示例实施例的导出已标记训练数据的方法的流程图,该已标记训练数据用于训练在基本SSE CatReco服务中使用的SSE模型。

[0018] 图9示出了根据另一示例实施例的用来训练用于基本SSE CatReco 服务的SSE模

型的流程图。

[0019] 图10示出了根据示例实施例的用于执行SSE统计语言建模 (SLM) - 梯度增强机器 (GBM) 运行时过程以生成CatReco的流程图。

[0020] 图11示出了根据示例实施例的用于执行SSE-SLM重新排序运行时过程的流程图。

[0021] 图12示出了根据一个示例实施例的用于执行SSE-SLM-GBM离线训练过程的第一部分的流程图。

[0022] 图13示出了根据一个示例实施例的用于执行SSE-SLM-GBM离线训练过程的第二部分的流程图。

[0023] 图14是示出根据一些示例实施例的可以安装在机器上的软件架构的示例的框图。

[0024] 图15示出了根据示例实施例的具有计算机系统的形式的机器的示图表示,在所述计算机系统中,可以执行一组指令以使所述机器执行本文讨论的方法中的任意一个或多个方法。

[0025] 图16示出了比较和识别发布的相关类别的示例方法。

[0026] 本文提供的标题仅为方便起见,而不一定影响所使用的术语的范围或含义。

具体实施方式

[0027] 以下描述包括体现本公开的示意性实施例的系统、方法、技术、指令序列和计算机程序产品。在下文的描述中,为了解释的目的,阐述了很多细节以提供对本发明主题的各种实施例的理解。然而,本领域技术人员将显而易见的是,本发明主题的实施例可以在没有这些具体细节的情况下实施。一般地,不必详细示出众所周知的指令实例、协议、结构和技術。

[0028] 在发布语料库中,建立了非常大规模的类别,以按照精细的粒度组织数十亿的不同发布(产品报价)。类别分类系统通常用于帮助卖方基于少量的标题关键词对发布列表进行分类。

[0029] 各种实施例描述了并行学习框架,以从无监督的用户日志中自动地导出极大规模的已标记数据(例如,数十亿),并将它们用于监督机器学习模型训练。

[0030] 示例实施例使用序列语义嵌入 (SSE) 方法将列表标题(例如,列出的发布的标题关键字)和类别树路径编码为语义向量表示,如<源序列,目标序列>对。源语义向量表示和目标语义向量表示的向量距离可被用作相似性度量,以获得分类回调候选集。分类回调候选集可以表示由LeafCat ID标识的类别树中的多个LeafCat。

[0031] 在其他实施例中,训练每个类别(例如,LeafCat)的语言模型,使得可以利用来自句子嵌入相似性分数(使用SSE建模导出)和语言模型复杂度分数(使用统计语言建模 (SLM) 导出)的梯度增强机器 (GBM) 整合信号对分类回调候选集进行重新排序。通过这种组合的SSE-SLM-GBM方法生成的类别推荐 (CatReco) 结果显得远远优于其他各种方法。例如,使用覆盖19000个以上的不同LeafCat的370,000个以上样本的基准测试结果显示出(超过生产基线 (production baseline) 的) 以下改进:系统响应时间快了10倍以上(例如,~200ms至~20ms),并且分类错误对于排名第1的CatReco减少了24.8%,对于排名前3的 CatReco减少了31.12%,且对于排名前10的CatReco减少了54.52%。

[0032] CatReco的准确性,特别是排名第1的推荐叶类别 (LeafCat) 的准确性可直接影响

用户(例如买方和/或卖方)的整体体验,因为有关发布的几个重要信息,例如卖方标签、列表费用和产品匹配对于发布而言依赖于LeafCat。此外,识别排名第1的推荐LeafCat的准确性通常是企业对消费者(B2C)自动分类流程的瓶颈。发布系统排名第1的CatReco 的准确性会对商品总量(GMV)产生直接影响,商品总量指示在特定时间范围内通过特定市场销售的商品的总销售美元价值。

[0033] 参考图1,示出了高级的基于客户端-服务器的网络架构100的示例实施例。具有基于网络的发布或支付系统的示例形式的联网系统102经由网络104(例如互联网或广域网(WAN))向一个或多个客户端设备 110提供服务器侧功能。图1示出了例如在客户端设备110上执行的网络客户端112(例如浏览器,比如由华盛顿州雷德蒙德的Microsoft公司开发的Internet Explorer®浏览器)、客户端应用114和编程客户端116。

[0034] 客户端设备110可以包括但不限于:移动电话、台式计算机、膝上型计算机、个人数字助理(PDA)、智能电话、平板计算机、超级本、上网本、笔记本电脑、多处理器系统、基于微处理器或可编程的消费电子产品、游戏机、机顶盒或用户可以用来访问联网系统102的任何其他通信设备。在一些实施例中,客户端设备110可以包括显示模块(未示出)以显示信息(例如,以用户接口的形式)。在另一些实施例中,客户端设备110可以包括触摸屏、加速度计、陀螺仪、相机、麦克风、全球定位系统(GPS)设备等中的一个或多个。客户端设备110可以是用于执行涉及联网系统102内的数字发布的交易的用户设备。在一个实施例中,联网系统102是基于网络的市场,其响应于对产品列表的请求,发布包括在基于网络的市场上的可用的产品的列表的公告,并且管理这些市场交易的支付。网络104的一个或多个部分可以是adhoc网络、内联网、外联网、虚拟专用网(VPN)、局域网(LAN)、无线LAN(WLAN)、WAN、无线WAN(WWAN)、城域网(MAN)、互联网的一部分、公共电话交换网(PSTN)的一部分、蜂窝电话网、无线网络、WiFi网络、WiMax网络、另一类型的网络或两个或更多个这样的网络的组合。

[0035] 客户端设备110中的每一个可以包括一个或多个应用(也称作“app”),例如但不限于web浏览器、消息传送应用、电子邮件(email)应用、发布系统应用(也称作市场应用)等。在一些实施例中,如果发布系统应用被包括在客户端设备110中指定的一个中,则该应用可以被配置为本地提供用户界面以及功能中的至少一些,其中该应用被配置为根据需要与联网系统102通信,以获得本地不可用的数据或处理能力(例如,访问可供销售的发布的数据库、认证用户、验证支付方法等)。相反,如果发布系统应用未被包括在客户端设备110中,则客户端设备110可以使用其web浏览器来访问联网系统102上容纳的发布系统(或其变型)。

[0036] 一个或多个用户106可以是人、机器或与客户端设备110交互的其他装置。在示例实施例中,用户106不是网络架构100的一部分,但可以经由客户端设备110或其它装置与网络架构100进行交互。例如,用户向客户端设备110提供输入(例如,触摸屏输入或字母数字输入),并且该输入经由网络104被传送到联网系统102。在这种情况下,联网系统102响应于从用户接收输入,经由网络104将信息传达到客户端设备 110以呈现给用户。以这种方式,用户可以使用客户端设备110与联网系统102交互。

[0037] 应用程序接口(API)服务器120和网络服务器122耦合至一个或多个应用服务器140,并分别向一个或多个应用服务器418提供编程接口和网络接口。应用服务器140可以主控(host)一个或多个发布系统 142和支付系统144,发布系统142和支付系统144中的每一个可以包括一个或多个模块或应用,并且该模块或应用中的每一个可以体现为硬件、软件、

固件或它们的任意组合。相应地,应用服务器140被示为耦合到一个或多个数据库服务器124,所述数据库服务器促进对一个或多个信息存储库或数据库126的访问。在示例实施例中,数据库126是存储要公告到发布系统120的信息(例如,发布或列清单)的存储设备。根据示例实施例,数据库126还可以存储数字发布信息。

[0038] 另外,在第三方服务器130上执行的第三方应用132被示为具有经由API服务器120提供的编程接口对联网系统102的编程访问。例如,第三方应用132利用从联网系统102获取的信息,支持第三方所拥有的 web系统上的一个或多个特征或功能。例如,第三方web系统提供由联网系统102的相关应用支持的一个或多个促销、市场或支付功能。

[0039] 发布应用142可以向访问联网系统102的用户提供多个发布功能和服务。支付系统144同样可以提供多个功能以执行或有助于支付和交易。虽然发布系统142和支付系统144在图1中都被示为形成联网系统102的一部分,但是应当理解,在备选实施例中,每个系统142和144可以形成与联网系统102分离且不同的支付服务的一部分。在一些实施例中,支付系统144可以形成发布系统142的一部分。

[0040] 列表系统150提供可操作以使用用户选择的数据执行列出待售的发布的各个方面的功能。在各种实施例中,卖方可以通过提供所列发布的标题或描述来列出发布(使用列表系统150)。标题可以被称为列表标题,并且由列表系统150(或发布系统142内的其他组件)使用来为列出的发布提供CatReco。在其他实施例中,列表系统150可以向数据库126、第三方服务器130、发布系统120和其它源访问用户选择的数据。在一些示例实施例中,列表系统150分析用户数据以执行用户偏好的个性化。随着更多内容被用户添加到类别,列表系统150可以进一步细化个性化。在一些示例实施例中,列表系统150与发布系统120(例如访问发布列表)和支付系统122进行通信。在备选实施例中,列表系统150是发布系统120的一部分。

[0041] 此外,虽然图1示出的基于客户端-服务器的网络架构100采用了客户端-服务器架构,但是本发明主题当然不限于此种架构,并且可以同样良好地应用于例如分布式或对等架构系统。各个发布系统142、支付系统144和列表系统150还可以被实现为独立的软件程序,所述独立的软件程序不一定具有联网能力。

[0042] web客户端112可以经由web服务器122所支持的web接口来访问各个发布系统142和支付系统144。类似地,编程客户端116通过API服务器120所提供的编程接口访问由发布系统142和支付系统144提供的各种服务和功能。例如,编程客户端116可以是销售者应用(例如,由加利福尼亚州圣何塞的eBay®公司开发的Turbo Lister应用),其用于使销售者能够以离线方式编写和管理联网系统102上的列表,并且执行编程客户端116与联网系统102之间的批处理模式通信。

[0043] 附加地,在第三方服务器130上执行的第三方应用132被示出为经由API服务器120提供的编程接口对联网系统102进行编程访问。例如,第三方应用132可以利用从联网系统102取回的信息来支持第三方容纳的web系统上的一个或多个特征或功能。例如,第三方web系统提供由联网系统102的相关应用支持的一个或多个促销、市场或支付功能。

[0044] 图2是根据示例实施例,更详细地示出图1的列表系统150的框图。这里,列表系统150包括列表服务器200,其用于执行与列表的发布有关的后端处理。除了其他组件之外,列表系统150还包括类别推荐(CatReco)组件202。用户可以直接使用用户设备204通过与列

表用户界面206交互来列出待售的发布,以提供发布的细节以供列出。列表用户界面206将该信息传送给列表服务器200。该过程本质上可以是交互式的。例如,用户通过列表用户界面206的某些输入被发送给列表服务器200,此时列表服务器200提供反馈,然后该反馈可以使用户改变或添加所提供的列表信息。

[0045] 出于本公开的目的,讨论将限于由CatReco组件202实现的列表服务器200的CatReco方面。在一个示例实施例中,用户可以经由列表用户界面206输入标题或其他文本输入,然后可以将其传递给CatReco组件202。然后,CatReco组件202可以为发布列表提供有序的建议类别列表,然后用户可以通过列表用户界面206从有序的建议类别列表中进行选择。在另一示例实施例中,用户(例如,B2C卖方)可以上载要由列表系统150列出的发布列表。发布列表包括与列表中每个条目相关联的列表标题和类别(基于卖方的分类)。然后,CatReco组件202可以自动将类别(基于卖方的分类)映射到针对每个条目的类别(基于发布系统142的分类)。卖方可以在卖方提供的库存列表(例如,具有列表标题和类别的条目)中提供卖方的分类,或者卖方可以提供卖方的分类的副本以便上载到发布系统142中。

[0046] CatReco组件202的各个实施例(与列表系统150和发布系统142内的其他组件相结合)利用SSE与SLM重新排序和GBM方法来为列出的发布的类别建立准确、稳健和快速的推荐。

[0047] 列表用户界面206可以采用许多形式。在一个示例实施例中,列表用户界面206是由用户设备204上的web浏览器执行的web页面。在另一示例实施例中,列表用户界面206是安装在移动设备上的移动应用。图3A和3B示出了由列表用户界面206生成的用户界面的示例,该用户界面用于列出发布,且用于为列出发布选择类别。

[0048] 列表服务器200还可以由第三方服务208经由列表API 210来访问。第三方服务208的示例是web系统,其通过代表卖方列出发布来在列出过程中主动帮助卖方。列表API 210可以被专门设计为与列表服务器202交互,并被分发给多个第三方服务208。

[0049] 一旦用户选择了列表的类别(至少部分地,由于CatReco组件202)或者列表系统自动将类别从卖方的分类映射到发布系统142的分类,列表服务器200发送该发布列表给库存管理服务器212,库存管理服务器212通过将列表存储在列表数据库214中来管理发布列表的过程。这可以通过分布式架构(例如Hadoop)来完成。

[0050] 然后,模型服务器216可以从列表数据库214获得关于列表的信息,以执行离线训练来创建和/或修改在向用户推荐类别时由CatReco组件202使用的模型(包括LeafCat模型)。如上所述,训练每个类别(例如,LeafCat)的语言模型,使得可以利用来自句子嵌入相似性分数(使用SSE建模导出)和语言模型复杂度分数(使用SLM导出)的梯度增强机器(GBM)整合信号对分类回调候选集进行重新排序。在各种实施例中,模型服务器216提供训练用于计算SSE-SLM-GBM CatReco结果的各种模型的功能。在一些实施例中,模型服务器216可以获得用于执行SSE模型的离线训练的信息。

[0051] 在各种实施例中,SSE用于将符号序列(如短语、句子或段落)编码成连续的维度向量空间,其中语义级的相似序列在该向量空间中具有更接近的表示。该SSE方法可以自动捕获列表标题的深层潜在语义含义,并将其语义级含义投射到共享的多维向量空间中。

[0052] 深度学习最近在自然语言处理(NLP)中表现出很大的希望。该领域的NLP研究人员正在尝试各种方式将符号序列(例如,短语、句子、段落和文档)编码到被称为语义空间的多

维向量空间。语义级的相似序列将在该多维空间中具有更接近的表示。在该领域中的研究导致采用句子而不仅仅是词的向量空间表示。通常，短语或句子而不是单个词更好地定义上下文信息。在各种实施例中，利用句子嵌入的研究来推荐卖方在发布系统上列出的发布的类别。

[0053] 在示例实施例中，使用SSE来嵌入给定列表标题的深层潜在语义含义并将其投射到共享语义向量空间。向量空间可被称为对象（被称为向量）的集合。向量空间可以通过它们的维度来表征，该维度指定空间中独立方向的数量。语义向量空间可以表示短语和句子，并且可以捕获针对NLP任务的语义。

[0054] 类似地，利用不同的投射函数，使用SSE来嵌入给定类别树路径的深层潜在语义含义（即，从顶层到叶级）并将其投射到共享语义向量空间。这种SSE方法使CatReco能够从列表标题中捕获上下文信息和深层语义含义，并且能够处理诸如同义词、拼写错误、复合词、分裂词等词中的大的差异。

[0055] 在示例实施例中，在系统运行时，将传入的列表标题投射到共享语义向量空间，并且列表系统推荐如下LeafCat：针对来自列表系统使用的类别分类的叶类别，该LeafCat具有与离线的预先计算的SSE列表的最接近的SSE表示。在另一示例实施例中，在系统运行时，将传入的列表标题投射到共享语义向量空间，并且列表系统推荐叶类别集合，该叶类别集合被用作列表系统中的其他服务的输入以生成CatReco结果和分数。例如，其他服务可以包括SLM重新排序服务或GBM融合预测服务。

[0056] 训练各种深度语义模型，以将语义相似的短语投射到彼此接近的向量，并将语义不同的短语投射到远离的向量。在训练时，如果列表标题 T 可以被分类为LeafCat C_1 ，那么 T 和 C_1 的投射语义向量空间值应尽可能接近，即 $||SSE(T) - SSE(C_1)||$ 应被最小化；而对于任何其他叶类别 C_n ，投射语义向量空间值应该尽可能地远，即 $||SSE(T) - SSE(C_n)||$ 应被最大化。在训练期间，可以计算语义向量之间的余弦相似性。因此，可以通过余弦相似性来测量两个向量之间的语义相关性。

[0057] 在各种实施例中，机器学习被用于最大化源 (X)（例如，列表标题）和目标 (Y)（类别树中的叶类别）之间的相似性以生成CatReco。SSE 模型可以基于深度神经网络 (DNN) 和/或卷积神经网络 (CNN)。DNN 是一种人工神经网络，在输入和输出层之间具有多个隐藏的单元层。DNN可以将深度学习架构应用于递归神经网络。CNN由一个或多个卷积层组成，顶部具有完全连接的层（例如与典型的人工神经网络匹配的层）。CNN还使用绑定权重和池化层。DNN和CNN都可以使用标准的反向传播算法进行训练。图7-9提供了用于训练SSE模型的示例流程图。训练的SSE模型在运行时期由基本SSE CatReco服务使用，如图6D 所示

[0058] SSE模型需要大量已标记数据以用于模型训练过程。通过手动标记过程获得大量已标记数据非常昂贵。这个限制可以通过使用利用数百万卖方的在线行为的并行学习方法自动导出干净的已标记训练数据来解决。可以使用两层过滤器来实现该并行学习方法以自动导出干净的训练数据（列表标题和叶类别的对）来用于SSE训练。图8示出了识别已标记训练数据对的示例方法。

[0059] 在示例实施例中，CatReco任务给出关键字集合（来自诸如卖方的用户提供的查询或列表标题），提供来自类别树的相关叶类别的有序列表，该类别树表示由发布系统使用的分类。基于卖方提供的给定关键字集合，发布系统推荐与给定关键字集相关的叶类别，以及

每个所推荐叶类别的顺序或分数的一些概念。CatReco通常是列表系统中消费者销售流程的前几步(即,用于列出发布)之一。

[0060] 根据各种示例实施例,如各种实施例中所述,SSE用于将列表标题分类为发布系统使用的类别。该类别可以表示来自发布系统使用的分类的类别树中的LeafCat(也称为类别节点)。这种方法可扩展、可靠且成本低。在列表系统中将SSE用于CatReco有很多好处。

[0061] 首先,通过针对基于SSE的CatReco的并行学习框架自动创建训练数据,降低了手动标记训练数据的成本。自动生成的训练数据基于并行学习框架,该并行学习框架利用来自发布系统的数百万卖方行为和其他离线可用信息,以确保标记的高准确性。

[0062] 其次,基于SSE的CatReco消除了对已知最近邻居(KNN)回调集的依赖性。例如,KNN回调集可被代之以替换为SSE回调集。

[0063] 第三,可以通过在子词/字符级而不是在词级训练SSE模型来解决词汇外(OV)问题。除了自然地处理复合词、分裂词、拼写错误等之外,这允许CatReco处理大量的词汇词。鉴于SSE正在编码整个序列上下文,在子词级的建模不会丢失上下文语义信息

[0064] 第四,CatReco系统能够在运行时提供快速响应,因为所有类别树路径(例如,16,000个叶类别)的所有语义空间向量表示可被事先离线预先计算,并且还可以将日志级高效K维度(KD)树算法应用于快速识别最匹配的类别树路径。

[0065] 最后,可以在几秒内直接计算任何可能的新LeafCat的语义空间向量表示,而无需重新训练任何模型。这使得基于SSE的CatReco系统具有很高的可扩展性,尤其是当类别树有许多更新时。

[0066] 在各种实施例中,发布系统使用的分类被表示在类别树中。在备选实施例中,可以使用其他分类结构。尽管上面的示例描述了由发布系统生成的CatReco,但是要理解,各种实施例可以在其他类型的在线系统中实现,并且不限于发布系统。

[0067] 在其他示例实施例中,SSE可以用于将源(X)映射到目标(Y)以用于其他NLP任务,并且不限于将列表标题(例如,源)映射到类别树(例如,目标)以识别一个或多个叶类别。下表列出了各种NLP任务以及相关源的源和目标的示例。在下面的表1中,源

[0068] 表1

	任务	源 (X)	目标 (Y)
[0069]	企业对消费者(B2C) 批量内建	卖方分类, 标题	类别
	PT 分类器	标题	产品类型
	前/后端分类	类别	产品类型
	产品化	列表	产品
	类别需求	查询	类别
	产品类型需求	查询	产品类型
	左手过滤	查询	相关属性集
	[0070]	语义搜索	查询

[0071] 图3A示出了根据示例实施例的用于列出发布的用户界面300。字段310是用于卖方提供列表的标题的文本字段。在图3A所示的示例中, 卖方提供标题“泰坦的冲突电影”以描述用于列出的发布。列表的标题通常是列表的一般性描述(并且可以包括对与发布相关的属性的描述)。发布系统可以使用标题来识别一个或多个相关类别, 以便在下方列出该发布。用户界面元素320向卖方呈现相关类别。在该特定实施例中, 向用户呈现前3个类别以选择他/她想要在下方列出发布的类别。对于示例实施例, 每个类别表示类别树中的类别“叶”。根据图3A, 卖方选择第一类别“DVD和电影>DVD和蓝光光盘”。图3B示出了“DVD和电影>DVD和蓝光光盘”类别中的在发布系统上的发布列表的示例。

[0072] 当SSE被应用于映射特定的<源, 目标>对时, SSE源模型和SSE 目标模型的参数被优化, 使得相关的<源, 目标>对具有更接近的向量表示距离。以下公式可用于计算最小距离。

[0073]
$$SrcMod^*, TgtMod^* = argmin_{k \text{ in all training pairs}} ||SrcVec^k - TgtVec^k||$$

[0074] 其中,

[0075] ScrSeq=源序列;

[0076] TgtSeq=目标序列;

[0077] SrcMod=源SSE模型;

[0078] TgtMod=目标SSE模型;

[0079] SrcVec=源序列的连续向量表示(也称为源的语义向量); 以及

[0080] TgtVec=目标序列的连续向量表示(也称为目标的语义向量)。

[0081] 源SSE模型将源序列编码为连续向量表示。目标SSE模型将目标序列编码为连续向量表示。在示例实施例中, 向量各自具有大约100个维度。

[0082] 图4示出了卖方提供的列表标题的示例400。图4中所示的列表标题410是“hello kitty T恤。”在此示例中，显示了三个维度。还示出了具有根节点453的类别树450的两个叶节点451和452。源SSE模型产生源(X) 420的语义向量。X由向量[0.1, 2.3, 3.0]表示，目标SSE模型产生目标(Y1和Y2) 430和440的语义向量。叶节点451的Y1“衣服、鞋子、配饰>女孩>T恤”由向量[0.1, 2.2, 3.0]表示，且叶节点452的Y2“衣服、鞋子、配饰>男孩>T恤”由向量[0.5, 2.6, 2.3]表示。根据向量中维度的值，在此示例中，基于源和目标的语义向量中的维度，列表标题“hello kitty T恤”似乎更接近叶节点451“衣服、鞋子、配饰>女孩>T恤”而不是叶节点452“衣服、鞋子、配饰>男孩>T恤”。图4中所示的示例是仅具有3个维度的非常简单的示例。

[0083] 在其在其他实施例中，可以使用任何数量的维度。在示例实施例中，语义向量的维度存储在KD树结构中。KD树结构可以被称为用于组织KD空间中的点的空间划分数据结构。KD树可用于执行最近邻居查找。因此，给定空间中的源点，可以使用最近邻居查找来识别到源点的最近点。

[0084] 图5A是根据示例实施例的流程图500，其示出了使列表标题与系统的类别分类相匹配的运行时分类过程。列表系统150针对类别使用的分类可以由类别树表示，并且类别树中的每个叶子可以表示类别。流程图500包括操作510、520、530和540。

[0085] 在操作510处，列表系统150接收发布的列表标题。在操作520处，SSE用于将列表标题映射到列表系统150用于列出发布的类别分类。在操作530处，识别至少一个相关类别。相关类别是从发布系统用于列出发布的类别分类中识别的。在操作540处，将至少一个识别出的相关类别提供给设备以呈现给用户。

[0086] 图5B是根据示例实施例的流程图501，其示出了使列表标题与系统的类别分类相匹配的运行时分类过程。使用SSE来识别叶类别(LeafCat)标识(ID)的回调集。流程图501包括操作510、520、535和545。在操作510处，列表系统150接收发布的列表标题。在操作520处，SSE用于将列表标题映射到列表系统150用于列出发布的类别分类。在操作535中，识别相关类别集合。相关类别是从发布系统用于列出发布的类别分类中识别的。相关类别可以表示收到的列表标题的前N个类别。例如，N=50。在操作545中，将LeafCat ID的回调集提供给列表系统150中的服务。例如，服务可以是SLM重新排序服务或GBM融合预测服务。

[0087] 图6A是进一步详细地示出操作520的流程图600，该操作使用SSE将列表标题映射到发布系统用于发布列表的类别分类。映射操作520包括操作610、620和630。

[0088] 在操作610处，检索预先计算(即，使用目标SSE模型)的目标(Y)的语义向量。预先计算的目标(Y)的语义向量创建语义向量空间。在示例实施例中，使用目标SSE模型计算目标系统的类别分类条目。预先计算的目标(Y)的语义向量是离线计算的，并且结合图6B来进一步详细描述。目标SSE模型将目标序列编码为连续向量表示。

[0089] 在操作620处，将源(X)的语义向量表示投射到共享语义向量空间中。将利用目标(Y)的语义向量创建的语义向量空间与源(X)的语义向量组合，以创建共享语义向量空间。源SSE模型用于创建列表标题的语义向量表示。

[0090] 在操作630处，识别(分类条目的)目标(Y)语义向量表示，该目标(Y)语义向量表示具有与共享语义向量空间内的(列表标题的)源(X)语义向量表示最接近的语义向量表示。分类条目可以表示LeafCat。在示例实施例中，使用余弦相似性函数来计算语义相关性sim

(X,Y)。结合图6C描述操作620和630的子操作的示例。

[0091] 如图6B所示,流程图601包括操作611-614。根据图6B,在操作 610处检索预先计算的目标(Y)的语义向量。

[0092] 在操作611处,访问目标。对于示例实施例,访问对列表系统150 的类别树的路径进行表示的目标。该路径表示LeafCat的根。对于示例实施例,从列表系统150的数据库访问类别树路径。

[0093] 在操作612处,对来自目标的类别树路径执行词散列。在示例实施例中,使用字母三元组(letter-trigram)来执行词散列。基于字母三元组的词散列采用原始短语(例如,根到叶的路径),进行预处理(例如,将 #添加到空白空间)并识别三字母(tri-letters)。词散列可用于创建大词汇表的紧凑表示。例如,500,000的词汇可被减少到30,000个字母三元组。词散列创建了对于拼写错误、词形变化、复合词、分裂词等都很健壮的列表系统150或其他系统。此外,看不见的词也可以使用词散列概括。

[0094] 在操作613处,目标SSE模型用于生成目标的语义向量(也称为语义向量表示)。

[0095] 在操作614处,将目标的语义向量存储在存储器设备614中的KD 树中。在示例实施例中,目标语义向量的维度存储在KD树结构中。在示例实施例中,目标语义向量表示类别树中的每个LeafCat的向量。叶类别可被表示为叶节点,例如图4中所示的叶节点。列表系统150的类别树的一个示例包括超过19,000个类别树路径(例如,从根到叶)。

[0096] 通过预先计算目标语义向量,可以非常快速地计算对表示列表标题的源语义向量进行映射的过程。在各种实施例中,在运行时间之前预先计算目标序列(如图6B中的操作601所示),并且在运行时期间计算源序列向量,然后在运行时期间与目标序列向量进行比较。图6C示出了流程图670,其将计算目标序列向量的离线过程与运行时过程组合以计算源序列向量,使得可以将列表标题映射(即,通过使用SSE模型并计算源和目标语义向量之间的语义相关性(例如,使用余弦函数))到列表系统150使用的类别分类。

[0097] 如图6C所示,流程图670包括离线操作601和运行时操作610、620和630。离线操作601如图6B中所示。运行时操作610、620和630 在图6A中示出。操作620(将源(X)的语义向量表示投射到共享语义向量空间中)包括子操作615和616。操作630(识别(分类条目的)目标(Y)语义向量表示,该目标(Y)语义向量表示具有与共享语义向量空间内的(列表标题的)源(X)语义向量表示最接近的语义向量表示)包括子操作617和618。

[0098] 如图6C所示,在操作601处离线计算目标的语义向量表示。在操作610处,检索预先计算的目标的语义向量。

[0099] 在操作615处,对源执行词散列。在示例实施例处,源表示列表的列表标题。在操作616处,使用源SSE模型生成源的语义向量。在示例实施例中,组合操作615和616被用于将源的语义向量表示投射到共享语义向量空间中(如操作620所示)。

[0100] 在操作617处,估计相关性相似性 $\text{sim}(X,Y)$ 。在操作618处,识别具有到X(表示为源语义向量)的最短距离的最佳匹配类别Y(表示为目标语义向量)。在示例实施例中,组合操作617和618被用于识别(分类条目的)目标(Y)语义向量表示,该目标(Y)语义向量表示具有与共享语义向量空间内的(列表标题的)源(X)语义向量表示最接近的语义向量表示(如操作630所示)。

[0101] 如上所述,在各种实施例中,可以通过学习源序列向量和目标序列向量之间的语

义相似性 $\text{sim}(X, Y)$ 来执行映射。在示例实施例中,语义相似性(也称为语义相关性)可以通过余弦相似性函数 $\text{sim}(X, Y)$ 来测量。在一些实施例中, X 表示源句子序列(即,从卖方的标题导出),且 Y 表示目标句子序列(即,从列表系统150的类别树导出)。余弦相似性函数的输出表示共享语义向量空间。通常, Y 的最佳匹配类别具有与 X 的最高相似性分数。源序列和目标序列表示已计算的向量序列,每个序列具有多个维度。

[0102] 图6D示出了流程图680,流程图680示出根据示例实施例的SSE运行时分类过程。图6D中所示的SSE运行时分类过程被用于通过将列表标题映射到列表系统150的类别分类来对列表标题进行分类。如上所述,流程图680可被用于通过将源映射到目标来执行多个任务,例如在上面的表1中所标识的那些任务。图6D中所示的基本SSE运行时分类过程也可以被称为基本SSE类别推荐(CatReco)服务680。基本SSE分类服务680表示用以获得回调集和相似性分数的SSE运行时解码过程。在示例实施例中,回调集表示类别树中排名前 N 的叶节点的集合。回调集和相似性分数可以由CatReco组件202(如图2所示)使用来生成SSE-SLM-GBM CatReco结果。下面结合图10描述SSE-SLM-GBM CatReco结果的生成。

[0103] 流程图680包括操作611-618和510。先前结合图6B描述了操作611-614。先前结合图6C描述了操作615-618。先前结合图5描述了操作510。

[0104] 操作611-614描述了用于计算存储在KD树结构中的目标的序列语义向量的离线过程。在运行时期访问KD树,使得源的序列语义向量可以被投射到具有目标的序列语义向量的共享语义向量空间中。估计相关性相似性 $\text{sim}(X, Y)$ (在操作617处),并且识别具有到 X 的最短距离的最佳匹配类别 Y (在操作618处)。最匹配的类别 Y 可以被称为类别树中与列表的列表标题匹配的排名第1的类别。在各种实施例中,识别前“ N ”个类别,使得可以向用户呈现 N 个类别中的多个。例如,在图3A中,排名前3的类别在用户界面300中被呈现给用户(例如,进行列出的卖方)。

[0105] 在示例实施例中,使用图7-9中描述的SSE模型训练过程来训练目标深度SSE模型613A和源深度SSE模型616A。

[0106] 在示例实施例中,CatReco任务用于将用户提供的列表标题分类到LeafCat。当存在大量类别时,对列表标题进行分类可能具有挑战性。CatReco任务经常被列表系统150的各种销售流程使用。对于示例实施例,列表系统150在美国可具有超过19,000个不同类别。列表系统通常致力于基于用户提供的关键字集合以及生成或向进行列出的卖方呈现CatReco时的响应时间来提高从超过19,000个类别中选择最相关类别的准确性。

[0107] 图7-9示出了根据示例实施例的由基本SSE CatReco服务680使用的SSE模型的训练过程。SSE运行时分类过程的实施例如图6D所示,并且当用于通过将列表标题映射到列表系统150的类别树路径来执行CatReco任务时,可以将其称为基本CatReco SSE服务580。图7示出了根据示例实施例的SSE训练模型的流程图700。图8示出了根据示例实施例的用于主动识别由SSE训练模型(图7中示出)使用的已标记训练数据对的流程图800。图9示出了SSE模型训练过程的示例,其包括图7和8中所示的各种操作和组件。

[0108] 参考图7,训练源SSE模型和目标SSE模型。操作710A、720A、730A和740A用于训练源SSE模型。操作710B、720B、730B和740B用于训练目标SSE模型。在操作701中,提供已标记训练数据对(列表标题、类别树路径)用于训练源SSE模型和目标SSE模型二者。在示例实施例中,使用图8中所示的流程图800来标识已标记训练数据对。

[0109] 在操作710A中,接收源列表标题(X)的原始句子序列。源列表标题(X)可以表示由进行列出的卖方提供的词序列。在操作720A中,对源列表标题(X)执行词散列。在存在非常大量的词汇词的情况下,对子词单元执行散列。在各种实施例中,执行字母三元组词散列,

[0110] 在示例实施例中,卷积层、最大池化层和语义层表示神经网络层。可以在那些神经网络层中配置多个节点(例如,如图9中所示的500个节点)。在其他实施例中,取决于数据大小,可以将节点的数量改变或配置为不同的数量。在操作730A中,使用卷积和最大池化(max-pooling)从源列表标题(X)中识别关键字和概念。

[0111] 在操作740A中,使用深度神经网络(DNN)提取源列表标题(X)的语义向量表示。DNN使用多于一个神经网络层将输入序列投射到语义向量空间中。

[0112] 在操作710B中,接收目标类别树路径(Y)的原始句子序列。对于示例实施例,列表系统150用于列出发布,并且可以包括超过19,000个类别树路径(Y)或CatLeaf。在操作720B中,可以在目标类别树路径(Y)上执行词散列。在存在非常大量的词汇词的情况下,对子词单元执行散列。在各种实施例中,执行字母三元组词散列,

[0113] 在示例实施例中,卷积层、最大池化层和语义层表示神经网络层。可以在那些神经网络层中配置多个节点(例如,如图9中所示的500个节点)。在其他实施例中,取决于数据大小,可以将节点的数量改变或配置为不同的数量。在操作730B中,使用卷积和最大池化从目标类别树路径(Y)识别关键字和概念。

[0114] 在操作740B中,使用深度神经网络(DNN)来提取目标类别树路径(Y)的语义向量表示。DNN使用多于一个神经网络层将输入序列投射到语义向量空间中。

[0115] 在操作750处,使用X和Y之间的语义向量距离来测量源列表标题(X)的语义向量表示与目标类别树路径(Y)的语义向量表示之间的相似性。在示例实施例中,通过余弦相似性来测量由函数 $\text{sim}(X,Y)$ 表示的语义相关性。

[0116] 当训练出源SSE模型和目标SSE模型时,可以使用目标SSE模型事先预先计算目标的所有类别分类条目的语义向量表示。另外,当需要从卖方映射任何新的发布列表时,可以将列表标题的语义向量表示投射到具有来自列表系统150的类别分类的类别树路径的语义向量表示的共享语义向量空间中。对于示例实施例,列表标题的正确映射将是具有与列表标题的语义向量表示最接近的语义向量表示的类别树路径。

[0117] 如上所述,当SSE被应用于映射特定的<源序列,目标序列>对时,SSE源模型和SSE目标模型的参数被优化,使得相关的<源,目标>对具有更接近的向量表示距离。以下公式可用于计算最小距离。

$$[0118] \quad \text{SrcMod}^*, \text{TgtMod}^* = \underset{k \text{ in all training pairs}}{\operatorname{argmin}} \quad \sum \quad \| \text{SrcVec}^k - \text{TgtVec}^k \|$$

[0119] 其中,

[0120] SrcSeq=源序列;

[0121] TgtSeq=目标序列;

[0122] SrcMod=源SSE模型;

[0123] TgtMod=目标SSE模型;

[0124] SrcVec=源序列的连续向量表示(也称为源的语义向量);以及

[0125] TgtVec=目标序列的连续向量表示(也称为目标的语义向量)。

[0126] 源SSE模型将源序列编码为连续向量表示。目标SSE模型将目标序列编码为连续向量表示。在示例实施例中，向量各自具有大约100个维度。

[0127] 已训练SSE模块用于实现运行时分类。在各种实施例中，SSE模型的训练是利用训练数据（例如，已标记训练数据对）离线执行的。在一些实施例中，自动导出已标记训练数据。对于示例实施例，每个已标记训练样本由<源序列表示，目标序列>对来表示。在示例实施例中，源序列表示发布列表的标题。目标序列通过列表系统150使用的类别分类的类别树路径来表示LeafCat。

[0128] 通常，良好的自然语言处理和机器学习方法需要已标记训练数据（即，监督学习）。使用数百万个已标记训练数据样本训练SSE模块可提高映射结果的准确性。在各种实施例中，使用列表系统150中的内建（onboarded）发布列表来训练SSE模型。已内建的现有发布列表使SSE模型能够使用相关数据进行快速训练。例如，位于加利福尼亚州圣何塞的eBay公司等公司可以访问数据仓库中记录的数十亿内建发布列表以及其卖方的库存分类信息。可以根据eBay之前的交易数据处理这些已内建发布列表，以挖掘、加入和过滤掉数百万这样的已标记训练数据。

[0129] 图8示出了根据示例实施例的导出已标记训练数据的方法的流程图800。在操作810处，访问存储在发布系统142的数据仓库中的来自于列表标题的历史数据。访问与卖方内建的先前发布列表相关的历史数据，该历史数据可以存储在来自发布系统142的数据仓库中。对于各种实施例，与先前发布列表相关的历史数据包括列表标题和进行列出的卖方在列出过程期间选择的类别。

[0130] 在操作820处，访问存储在发布系统的数据库中的类别树的LeafCat。在示例实施例中，LeafCat可以包括超过19,000个条目。

[0131] 基于列表系统150的类别分类，训练包括列表标题和LeafCat的类别树路径的数据。

[0132] 在操作830处，在某个时间段（例如，每八周），针对每个LeafCat识别列表标题。通过操作810和820访问的数据被用于针对每个叶类别识别列表标题。

[0133] 然后，通过在操作840处应用过滤器A并且在操作850处应用过滤器B来过滤训练数据。通过使用过滤器A和B，列表系统150检查卖方对于类别的选择是否与来自列表系统150的CatReco组件202的第一推荐相匹配。如果存在匹配，则列表系统150检查列表误分类（miscat）分数是否低。低分经常表明列表发布被误分类为错误的LeafCat中。低分的示例可以是50。如果列表发布通过了过滤器A和B两者，则该（列表标题，类别树路径）对被视为干净的训练样本。

[0134] 在操作860处，识别已标记训练数据对（列表标题，类别树路径）。在各种实施例中，这是识别已标记训练对的自动过程，已标记训练对被训练过程用于实现主动学习。流程图800中示出的方法可以是自动的，使得已标记训练数据对被定期识别并且被用于通过机器学习主动训练SSE模型过程，如图9中的流程图900所示。

[0135] 图9示出了SSE模型训练过程并且在示例实施例中组合了流程图700（图7中示出）和流程图800（图8中示出）。SSE模型训练过程的一个重要目标是尝试获得优化的源SSE模型和优化的目标SSE模型，使得对于所有训练样本对，源序列的连续向量表示与目标序列的连续向量表示之间的距离被最小化。在各种实施例中，机器学习被用于优化源SSE模型和目

标SSE模型以实现最小化该距离的目标。

[0136] 图9示出了根据示例实施例的流程图900。图9所示的方法包括操作710A-740A、710B-710B和750 (其被结合图7描述) 以及操作810-860 (其被结合图8描述)。

[0137] 根据图9,干净的训练对(列表标题,类别树路径)被用于训练源SSE模型和目标SSE模型。训练对可以被称为干净的训练对,因为生成训练对的过程使用过滤器A和B过滤掉了错误分类的对(在操作840和850处)。在一个示例中,列表标题是“视频监视器,摩托罗拉-无线视频婴儿监视器-白色”,类别树路径是“婴儿>婴儿安全和健康>婴儿监视器”。来自训练对的列表标题作为输入被提供到源SSE模型中,并且来自训练对的类别树路径作为输入被提供到目标SSE模型中。在示例实施例中,通过余弦相似性测量(训练数据中的列表标题的源语义向量和类别树路径的目标语义向量)的语义相关性被称为相似性分数。在示例实施例中,CatReco组件202内的机器学习系统使用干净的训练对(在操作860处识别)来训练源SSE模型和目标SSE模型。

[0138] 对于示例实施例,由基本SSE CatReco服务680(如图6D所示)提供的基本SSE运行时分类过程利用使用SSE模型训练过程训练的目标深度SSE模型613A和使用SSE模型训练过程900(如图9所示)训练的源深度SSE模型616A。

[0139] 尽管图9中所示的SSE模型训练过程示出了使用已标记训练对(列表标题,类别树路径)训练源和目标SSE模型,其中列表标题为“视频监视器,摩托罗拉-无线视频婴儿监视器-白色”,并且类别树路径是“宝贝>婴儿安全与健康>婴儿监视器”,但是图9所示的SSE模型训练过程可用于训练其他类型的已标记训练对。例如,当执行其他类型的任务(例如上面表1中所示的那些任务)时,可以使用标记训练对(列表标题,产品类型树路径)或已标记训练对(分类树路径,产品类型树路径)。

[0140] 在各种实施例中,列表系统150(图2中所示)的CatReco组件202可以结合使用由SLM重新排序服务1110和梯度增强机器(GBM)融合预测服务1030提供的统计语言建模(SLM)来利用基本SSE CatReco服务680。在示例实施例中,SLM重新排序服务1110和GBM融合预测服务1030也可以由列表系统150执行。根据示例实施例的SSE-SLM-GBM方法的高级框图在图10中示出。图10中所示的流程图1000示出了用于生成具有分数的SSE-SLM-GBM CatReco结果的过程。

[0141] 在各种示例实施例中,SLM被用于提高CatReco组件202提供的推荐的准确性。SLM是数据驱动的建模方法,其试图定性给定文本输入(例如句子、列表标题、或搜索查询)的可能性。SLM能够利用大量无监督的文本数据(例如,未被标记且因此没有明显的结构的文本数据)。在一个示例中,SLM用于基于无监督的列表标题训练针对每个LeafCat的语言模型,且然后使用适当的LeafCat的语言模型评估新列表标题的句子对数概率(SLP)。这可以针对每个候选LeafCat重复进行。在各种实施例中,在基本SSE CatReco服务680已生成相似性分数和SSE回调集之后执行用于所建议类别的排序的重新排序过程。回调集可表示基本SSE CatReco服务680产生的前N个类别。

[0142] 值得注意的是,在示例实施例中,使用SLM重新排序服务1110仅评估在前N个叶类别中列出的类别(由基本SSE CatReco服务680标识)。这比在所有可能的类别(例如,超过19,000个叶类别)上运行SLM算法明显更有效。

[0143] 另外,在示例实施例中,GBM用于组合若干估计器的预测,以便进一步细化所建议

的类别,将各种分数和数据融合在一起,如下所述。

[0144] 根据图10,在操作1001处接收发布列表的标题。将发布列表的标题提供给基本SSE CatReco服务680和SLM重新排序服务1110。

[0145] 基本SSE CatReco服务680被用于识别前N个LeafCat,其由列表标题的LeafCat ID的SSE回调集1010限定。LeafCat ID 1010的SSE回调集作为输入被提供到SLM重新排序服务1110中。在示例实施例中,SLM重新排序服务1110包括两个组件:SLM运行时分类阶段1110A(图11中示出)和SLM训练阶段1110B(图12中示出)。

[0146] 不是在输入文本字符串上使用k最近邻居(KNN)算法(例如,表示发布列表的标题)来识别叶类别集合,而是使用基本SSE CatReco服务680来识别叶类别的集合(即,前N个LeafCat)。基于对输入文本串执行的SLM算法,针对每个LeafCat的组合SLM 1232、针对每个LeafCat的对数似然概率(LLP) 1212、针对每个LeafCat的预期复杂度和标准偏差(也称为预期PPL和PPL_Std) 1236来(由SLM重新排序服务1110)重新排序叶类别的集合(由LeafCat ID 1010的SSE回调集限定)。将结合图11更详细地讨论LLP和PPL。

[0147] 在操作1030处,GBM融合预测服务1030接收SSE回调集LeafCat ID 1010、针对每个LeafCat的LLP 1212、每个LeafCat的预期PPL和PPL Std 1236、以及来自SLM重新排序服务1110的输出(即重新排序的LeafCat集合)来作为输入。然后在操作1030处,使用GBM融合预测服务1030融合所接收的各种输入以计算具有对应分数的所推荐LeafCat的有序列表。在在1040处显示GBM融合预测的结果。

[0148] 图11是示出根据示例实施例的SLM重新排序服务1110的SLM运行时分类阶段1110A的图。

[0149] 根据图11,输入列表标题1001被提供给基本SSE CatReco服务680。基本CatReco服务680生成LeafCat标识(ID) 1010的SSE回调集,其被作为输入提供给SLM运行时分类阶段1110A。

[0150] 针对每个LeafCat的LLP 1212、针对每个LeafCat的组合SLM 1232以及针对每个LeafCat的预期PPL和PPL_Std 1236由SLM运行时分类阶段1110A访问。更具体地,针对每个LeafCat的LLP 1212是离线预先计算的并存储在文件中,并在运行时被加载到存储器中。针对每个LeafCat的组合SLM 1234是针对每个LeafCat的SLM模型,它被离线预训练并在运行时加载到存储器中。针对每个LeafCat的预期PPL和PPL STD1236也在模型训练过程中被预先离线计算并保存到文件中,并在运行时加载到存储器中。结合图12进一步详细描述针对每个LeafCat的LLP 1212、针对每个LeafCat的组合SLM 1232以及针对每个LeafCat的预期PPL和PPL_Std 1236的预先计算。

[0151] 在SLM运行时分类阶段1110A处,计算深信号以测量给定列表偏离指派的叶类别有多远。假设运行时发布列表标题为T,卖方将其置于类别C下,并且将发布的运行时复杂度计算为PP(T),其偏差信号计算为:

$$[0152] \quad Deviation_PP(C,T) = \frac{PP(T)}{Mean_PP(C) + \alpha * STD_PP(C)}$$

[0153] 其中 α 是可以微调的参数(在示例实施例中,它被设置为2.0)。

[0154] 最后,Mean_PP(C)、STD_PP(C)、PP(T)和Deviation_PP(C,T)可以作为深特征与传统的浅特征(例如价格、状况、CatReco分数等)一同馈入GBM模型,以产生整合模型。

[0155] 在操作1120处,基于针对LeafCat的LLP 1212来识别针对候选 LeafCat ID的LLP。候选LeafCat ID基于LeafCat ID的SSE回调集。

[0156] 在操作1130处,基于针对每个LeafCat的组合SLM 1234来识别针对候选LeafCat ID的SLP。候选LeafCat ID基于LeafCat ID的SSE回调集。

[0157] 操作1120的输出(即,针对候选LeafCat ID的识别出的LLP)和操作1130的输出(即,针对候选LeafCat ID的识别出的SLP)被用作输入以在操作1140处计算SLM排序分数。SLM排序分数被用作操作1150的输入。在操作1150处,基于SLM排序分数计算SLM投票分数。在操作1150处,产生列表标题的SLM排序分数。

[0158] 在示例实施例中,通过将(加权的)各个SLP分数和LPP分数加在一起来计算针对每个LeafCat的SLM排序分数(SRS),例如通过使用公式 $SRS = SLP + 1.8 * LPP$ 。在示例实施例中,通过将1除以针对叶类别的最大SRS分数和各个SRS分数之间的差与1的总和来计算SLM投票分数,例如通过使用以下公式: $SLM投票分数 = 1 / (1 + Max_SRS - SRS)$ 。

[0159] 在操作1160处,将针对候选LeafCat ID的识别出的SLP和针对来自LeafCat ID的SSC回调集的LeafCat的预期PPL和PPL_Std用作输入,以在操作1160处计算SLM PPL偏差百分位数。在操作1160中,产生列表标题的SLM复杂度偏差信号。复杂度偏差信号可以称为深特征。在示例实施例中, $SLM PPL偏差百分位数 = CurPPL / (PPL_Mean + 2 * PPL_Std)$ 。CurPPL是指当前的复杂度,并在运行时计算。CurPPL是指针对候选LeafCat的SLM模型输入的新列表标题的PPL值。参考下面提供的公式,术语“PPL_Mean”可以被称为mean_PPL,术语“PPL_Std”可以被称为STD_PP。

[0160] 在SLM运行时重新排序阶段1110A期间,当SSE产生候选LeafCat Id的回调集时,在运行时计算出SLP、PPL和PPL_Deviation值,该SLP、PPL和PPL_Deviation值基于所请求的发布列表的标题,且针对每个候选LeafCat的针对LeafCat的对应组合SLM 1232。LLP、PPL、SLP、PPL_Deviation值被用于重新排名整个回调leafCat候选集。

[0161] 在示例实施例中,句子PPL可以如下计算。假设句子S由诸如 $\{w_1, w_2, \dots, w_N\}$ 的N个词的序列组成。计算S的复杂度:

$$[0162] \quad PP(S) = P(w_1 \dots w_N)^{-1/N} = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

[0163] 对于给定的LeafCat C,可能有M个句子(来自列表标题)作为调整集。它们可以被标示为 S_1, S_2, \dots, S_M 。对于这些标题句中的每一个,可以基于上面的公式计算其对应的复杂度。然后,可以根据以下公式找到针对给定LeafCat的预期复杂度值和相关标准偏差值(请注意,所有 mean_PP和STD_PP值都可以预先计算并存储以供运行时使用):

$$[0164] \quad Mean_PP(C) = Mean_PP(S_1 \dots S_M) = \frac{\sum PP(S_i)}{M}$$

$$[0165] \quad STD_PP(C) = STD_PP(S_1 \dots S_M) = \sqrt{\frac{\sum (PP(S_i) - Mean_PP(C))^2}{M - 1}}$$

[0166] 图12是示出根据示例实施例的SLM训练阶段1110B的图。对于示例实施例,SLM训练

阶段1110B是SLM重新排序服务1110的一部分。SLM训练阶段1110B访问包含发布信息的数据库1202,该发布信息可包括列表标题、搜索查询、产品名称等的信息。可以对该数据库执行各种搜索,以识别与正在为其创建SLM模型的特定LeafCat相关的信息。

[0167] 这里,已经指定了四个搜索:(1)在操作1204处,在最近X时间段(例如,8周)中针对LeafCat的列表的数量;(2)在操作1206处,LeafCat中每个发布的产品名称;(3)在操作1208处,在最近的X时间段中对LeafCat执行的查询;以及(4)在操作1210处,在最近的X时间段中针对LeafCat的列表标题。这些搜索中的每一个的结果以不同的方式使用。对于在操作1204处访问的最近X时间段中针对LeafCat的列表的数量,该信息用于在操作1212处为LeafCat创建对数先验概率(LPP)。该过程将在下面更详细地描述。

[0168] 对于在操作1206处访问的LeafCat中的每个发布的产品名称,在操作1214处,通过针对语料库的文本标准化,首先对该信息进行标准化(例如,纠正拼写错误或替代拼写),然后该信息用于为与叶类别的结构化数据相对应的结构化数据1216构建SLM。

[0169] 对于在操作1208处访问的最近X时间段中对LeafCat执行的查询,在操作1218处通过针对语料库的文本标准化首先对该信息进行标准化(例如,纠正拼写错误或替代拼写),然后使用该信息为LeafCat 1220构建SLM。

[0170] 在操作1210处访问最近X时间段中针对LeafCat的列表标题。该信息首先通过过滤器,包括过滤器A 1222和过滤器B 1224。这些过滤器1222、1224用于将列表标题缩减到最相关。这里,例如,过滤器A1222 识别以下列表:在该列表中,卖方类别选择与列表的排名最高的CatReco 匹配(基于分类算法)。例如,过滤器B 1224通过将每个列表的错误分类分数与阈值(例如,100个中的60个,其中300是列表被错误分类的最高可能性)进行比较来识别具有较低概率被错误分类的列表。在这方面,该过程在某种程度上是递归的,因为使用针对叶类别的SLM重新排序服务1110的运行过程导出错误分类分数,其在图12所示的该阶段中被训练。然后可以执行操作1226处的语料库的文本标准化以标准化已过滤结果的文本。这种标准化的结果可以通过两种方式来使用。首先,可以创建针对每个LeafCat标题1228的SLM作为训练集的一部分。另外,结果的其余部分可用在调整集中。

[0171] 然后,可以在操作1230处内插针对结构化数据的SLM(对应于叶类别的结构化数据)1216、针对每个LeafCat的SLM 1220以及针对每个 LeafCat的训练SLM 1228,以创建针对每个LeafCat的组合SLM 1232。

[0172] 在调整集侧,在操作1234处,LeafCat的组合SLM 1232以及语料库的文本标准化输出可以在操作1234处用于针对LeafCat的每个列表的 PPL和PPL_Std评估,以创建针对每个LeafCat标题的预期PPL和 PPL_Std 1236。对每个叶类别重复该过程。

[0173] 图13是示出根据示例实施例的GBM训练模型过程1300的图。在离线无监督GBM训练模型过程1300中,可以通过检查如何选择CatReco 以及相关的错误分类分数来以无监督的方式导出自举的已标记训练数据 1320的集合。一旦获得了已标记训练数据1320,就可以基于来自SLM 重新排序服务1110和基本SSE CatReco服务680的输出来准备GBM特征输入文件1360。更具体地,SLM重新排序服务1110产生针对训练数据的SLM复杂度偏差信号1330和针对训练数据的SLM排序分数1340,并且基本SSE CatReco服务680产生针对训练数据的SSE相似性分数 1350。然后,GBM训练过程可用于训练GBM模型。在操作1370处, GBM特征文件在操作1370处被用于GBM训练。GBM训练通过元数据产生GBM模型1380。

[0174] 根据图13,使用操作1302、1304、1306和1308获得已标记训练数据1320。在操作1302处,访问来自数据库1301的针对每个LeafCat的最近X时间段的列表标题。例如,最近的X时间段可以参考示例实施例中的最近8周。然后将两层过滤器应用于此信息。在操作1304处,过滤器A将消耗操作1302的输出并且仅保持具有与根据CatReco算法的排名最高的选择相匹配的卖方的类别选择的列表,然后将其结果传递给下一操作1306,过滤过滤器B的下一步骤,过滤器B仅保留具有小于第二预定阈值(例如,100个中的35个,意味着列表被错误分类的较低可能性)的误分类分数的列表。满足这两层过滤器A和B的要求的列表标题在操作1308处被标记为非误分类。

[0175] 如上所述,可以针对每条已标记训练数据1320从SLM重新排序服务1110导出复杂度偏差信号1330和SLM排序分数1340。另外,可以针对每条已标记训练数据1320从基本CatReco服务680导出SSE相似性分数1350。

[0176] 模块、组件和逻辑

[0177] 某些实施例在本文中被描述为包括逻辑或多个组件、模块或机构。模块可以构成软件模块(例如,机器可读介质上体现的代码)或硬件模块。“硬件模块”是能够执行某些操作的有形单元,并且可以按照某种物理方式配置或布置。在各种示例实施例中,一个或多个计算机系统(例如独立的计算机系统、客户端计算机系统或服务器计算机系统)或者计算机系统的一个或多个硬件模块(例如处理器或处理器组)可由元件(例如应用或应用部分)配置为操作为执行本文描述的特定操作的硬件模块。

[0178] 在一些实施例中,硬件模块可以按照机械方式、电子方式或其任意适当组合来实现。例如,硬件模块可以包括永久地被配置为执行特定操作的专用电路或逻辑。例如,硬件模块可以是专用处理器,如现场可编程门阵列(FPGA)或专用集成电路(ASIC)。硬件模块还可以包括由软件临时配置为执行特定操作的可编程逻辑或电路。例如,硬件模块可以包括由通用处理器或其他可编程处理器执行的软件。一旦由这样的软件配置,硬件模块就变成特定的机器(或机器的特定组件),其被专门定制用于执行所配置的功能,而不再是通用处理器。应理解:以机械方式、以专用和永久配置的电路或以临时配置的电路(例如由软件配置)实现硬件模块的决定可出于成本和时间的考虑。

[0179] 因此,短语“硬件模块”应理解为涵盖有形实体,是在物理上构造、永久配置(例如硬线连接)或临时配置(例如编程)为以特定方式操作或执行本文描述的特定操作的实体。如本文所使用的,“硬件实现的模块”指硬件模块。考虑临时配置(例如编程)硬件模块的实施例,无需在任一时刻配置或实例化硬件模块中的每一个。例如,在硬件模块包括被软件配置成为专用处理器的通用处理器的情况下,通用处理器可以在不同时间被配置为分别不同的专用处理器(例如包括不同的硬件模块)。因此,软件将特定的一个或多个处理器例如配置为在一个时刻构成特定硬件模块并在另一时刻构成不同的硬件模块。

[0180] 硬件模块可以向其他硬件模块提供信息并从其他硬件模块接收信息。因此,所描述的硬件模块可以被看作通信地耦合。如果同时存在多个硬件模块,则可以通过两个或更多个硬件模块之间的信号传输(例如通过适当的电路和总线)实现通信。在多个硬件模块在不同时间配置或实例化的实施例中,可以例如通过存储并获取多个硬件模块可访问的存储器结构中的信息来实现这样的硬件模块之间的通信。例如,一个硬件模块可以执行操作并在与其通信耦合的存储设备中存储该操作的输出。另一硬件模块接着可以稍后访问存储器

设备,以取回并处理所存储的输出。硬件模块还可以发起与输入或输出设备的通信,并且能够对资源(例如信息的集合)进行操作。

[0181] 此处描述的示例方法的各种操作可以至少部分地由临时配置(例如通过软件)或永久配置为执行相关操作的一个或多个处理器执行。无论是临时还是永久配置,这样的处理器可以构成操作以执行本文描述的一个或多个操作或功能的处理器实现的模块。如本文所使用的,“处理器实现的模块”指使用一个或多个处理器实现的硬件模块。

[0182] 类似地,本文描述的方法可以至少部分地由处理器实现,其中特定处理器或多个处理器是硬件的示例。例如,方法的至少一些操作可由一个或多个处理器或处理器实现的模块执行。此外,一个或多个处理器还可操作以支持在“云计算”环境中或作为“软件即服务”(SaaS)执行有关操作。例如,操作中的至少一些可由计算机(作为包括处理器的机器的示例)组执行,这些操作可经由网络(例如互联网)并经由一个或多个适当接口(例如应用接口(API))来访问。

[0183] 某些操作的执行可以分布在处理器中,并不只驻留在单个机器内,而是部署在多个机器中。在一些示例实施例中,处理器或处理器实现的模块可以位于单个地理位置(例如,在家庭环境、办公环境或服务器群中)。在其他示例实施例中,处理器或处理器实现的模块可以分布在多个地理位置中。

[0184] 机器和软件架构

[0185] 在一些实施例中,结合图1-6描述的模块、方法、应用等在机器和相关联的软件架构的上下文中实现。以下部分描述了适用于与所公开的实施例一起使用的代表性软件架构和机器(例如,硬件)架构。

[0186] 软件架构与硬件架构一起使用,以创建针对特定用途定制的设备 and 机器。例如,与特定软件架构耦合的特定硬件架构将创建移动设备,诸如移动电话、平板设备等。稍微不同的硬件和软件架构可以生成用于“物联网”的智能设备。而另一组合产生了在云计算架构中使用的服务器计算机。本文并没有介绍这样的软件和硬件架构的所有组合,因为本领域技术人员可以容易地理解在不同于本文所包含的公开内容的不同上下文中如何实现本发明。

[0187] 软件架构

[0188] 图14是示出代表性软件架构1402的框图1400,该代表性软件架构 1402可以结合本文所描述的各种硬件架构一起使用。图14仅是软件架构的非限制性示例,且应该了解,可以实施许多其他架构以促进实现本文中所描述的功能。软件架构1402可以在诸如图15的机器1500的硬件上执行,所述机器1500包括处理器1510、存储器1530和I/O组件1550。代表性的硬件层1404被示出,并且可以表示例如图15的机器1500。代表性的硬件层1404包括具有关联的可执行指令1408的一个或多个处理单元1406。可执行指令1408表示软件架构1402的可执行指令,包括图 1至图13的方法、模块等的实现。硬件层1404还包括存储器或存储模块1410,其也具有可执行指令1408。硬件层1404还可以包括如1412所示的其他硬件,其表示硬件层1404的任何其他硬件,诸如作为机器 1500的一部分示出的其他硬件。

[0189] 在图14的示例性架构中,软件1402可被概念化为层的堆栈,其中每层可提供特定的功能。例如,软件1402可以包括诸如操作系统1414、库1416、框架/中间件1418、应用1420和表示层1444等层。在操作上,层中的应用1420或其他组件可以通过软件栈调用API调用1424并接收被示出为响应于API调用1424的消息1426的响应、返回值等。所示出的层在本质

上具有代表性,并不是所有的软件架构都具有所有层。例如,一些移动或专用操作系统可能不提供框架/中间件层1418,而其他系统可以提供这样的层。其它软件架构可以包括附加层或不同层。

[0190] 操作系统1414可以管理硬件资源并提供公共服务。操作系统1414 可以包括例如内核1428、服务1430和驱动1432。内核1428可以用作硬件和其他软件层之间的抽象层。例如,内核1428可以负责存储器管理、处理器管理(例如调度)、组件管理、联网、安全设置等。服务1430可以为其它软件层提供其它公共服务。驱动1432可以负责控制底层硬件或与底层硬件接口连接。例如,取决于硬件配置,驱动1432可以包括显示器驱动、相机驱动、蓝牙®驱动、闪存驱动、串行通信驱动(例如通用串行总线(USB)驱动),Wi-Fi®驱动、音频驱动、电源管理驱动等等。

[0191] 库1416可以提供可由应用1420和/或其它组件和/或层利用的公共基础设施。库1416通常提供允许其他软件模块以比直接与底层操作系统 1414的功能(例如,内核1428、服务1430或驱动1432)接口连接更容易的方式执行任务。库1416可以包括可以提供诸如存储器分配功能、串操纵功能、数学功能等功能的系统1434库(例如,C标准库)。另外,库1416可以包括API库1436,例如媒体库(例如,用于支持各种媒体格式(诸如MPREG4、H.264、MP3、AAC、AMR、JPG、PNG)的呈现和操纵的库,)、图形库(例如,可以用于在显示器上渲染图形内容中的2D和3D的OpenGL框架)、数据库(例如,可以提供各种关系数据库功能的SQLite)、web库(例如,可以提供网络浏览功能的WebKit)等。库1416还可以包括各种各样的其它库1438,以提供到应用1420和其它软件组件/模块的许多其他API。

[0192] 框架1418(有时也称为中间件)可以提供可由应用1420或其他软件组件/模块使用的更高级别的公共基础设施。例如,框架1418可以提供各种图形用户界面(GUI)功能、高级资源管理、高级位置服务等。框架1418可以提供可以由应用1420和/或其它软件组件/模块利用的广泛范围的其它API,其中一些可以特定于特定操作系统或平台。

[0193] 应用1420包括内置应用1440和/或第三方应用1442。代表性的内置应用1440的示例可以包括但不限于联系人应用、浏览器应用、书籍阅读器应用、位置应用、媒体应用、消息传递应用和/或游戏应用。第三方应用1442可以包括任何内置应用以及各种其他应用。在具体示例中,第三方应用1442(例如,由除了特定平台的供应商之外的实体使用Android™或iOS™软件开发工具包(SDK)开发的应用)可以是在移动操作系统(例如iOS™、Android™、Windows®电话或其他移动操作系统)上运行的移动软件。在该示例中,第三方应用1442可以调用由诸如操作系统 1414之类的移动操作系统提供的API调用1424,以有助于实现本文描述的功能。

[0194] 应用1420可以利用内置操作系统功能(例如,内核1428、服务1430 和/或驱动1432)、库(例如,系统1434、API 1436和其他库1438)和框架/中间件1418来创建用户接口以与系统的用户交互。备选地或附加地,在一些系统中,与用户的交互可以通过表示层(诸如表示层1444)发生。在这些系统中,应用/模块“逻辑”可以与和用户交互的应用/模块的各方面分离。

[0195] 一些软件架构利用虚拟机。在图14的示例中,这由虚拟机1448示出。虚拟机创建软件环境,在该软件环境中应用/模块可以像在硬件机器(诸如图15的机器)上执行一样执行。虚拟机由主操作系统(图15中的操作系统1414)容纳,并且通常(尽管并不总是)具有管

理虚拟机的操作以及与主操作系统(即,操作系统1414)连接的接口的虚拟机监视器1446。软件架构在虚拟机(例如操作系统1450、库1452、框架/中间件1454、应用1456和/或呈现层1458)内执行。在虚拟机1448内执行的这些软件架构的层可以与先前描述的对应层相同,或者可以不同。

[0196] 示例机器架构和机器可读介质

[0197] 图15是示出了根据一些示例实施例的能够从机器可读介质(例如,机器可读存储介质)中读取指令并执行本文所讨论的方法中的任何一个或多个的机器1500的组件的框图。具体地,图15示出了计算机系统的示例形式的机器1500的示意性表示,在机器1500中,可以执行指令1516(例如,软件、程序、应用、小程序、app或其他可执行代码)以使机器1500执行本文所讨论的方法中的任何一个或多个。例如,指令可以使机器执行图14的流程图。附加地或备选地,这些指令可以实现图5A-13所描述的模块等等。指令将通用的未编程的机器转换成被编程为以所描述的方式执行所描述和示出的功能的特定机器。在备选实施例中,机器1500作为独立设备操作或可以耦合(例如,联网)到其他机器。在联网部署中,机器1500可以在服务器-客户端网络环境中以服务器机器或客户端机器的容量操作,或者作为对等(或分布式)网络环境中的对等机器操作。机器1500可以包括但不限于服务器计算机、客户端计算机、个人计算机(PC)、平板计算机、膝上型计算机、上网本、机顶盒(STB)、PDA、娱乐媒体系统、蜂窝电话、智能电话、移动设备、可穿戴设备(例如智能手表)、智能家居设备(例如智能家电)、其他智能设备、网络设备、网络路由器、网络交换机、网桥、或能够顺序地或以其他方式执行指定机器1500要采取的动作的指令1516的任何机器。此外,尽管仅示出了单个机器1500,但是术语“机器”也将被认为包括机器1500的集合,其单独地或联合地执行指令1516以执行本文讨论的方法中的任何一个或多个。

[0198] 机器1500可以包括可被配置为诸如经由总线1502彼此通信的处理器1510、存储器1530和I/O组件1550。在示例实施例中,处理器1510(例如,中央处理单元(CPU)、精简指令集计算(RISC)处理器、复杂指令集计算(CISC)处理器、图形处理单元(GPU)、数字信号处理器(DSP)、ASIC、射频集成电路(RFIC)、另一处理器或其任何适当组合)可以包括例如可以执行指令1516的处理器1512和处理器1514。术语“处理器”旨在包括可以包括可以同时执行指令的两个或更多个独立处理器(有时称为“核”)的多核处理器。尽管图15示出了多个处理器,但是机器1500可以包括具有单个核的单个处理器、具有多个核的单个处理器(例如,多核处理器)、具有单个核的多个处理器、具有多个核的多个处理器或其任何组合。

[0199] 存储器/存储装置1530可以包括存储器1532(比如,主存储器或其它存储存储设备)、以及存储单元1536,存储器器1532和存储单元1536两者都可例如经由总线1502由处理器1510访问。存储单元1536和存储器1532存储体现本文所述的任何一种或多种方法或功能的指令1516。在机器1500执行指令1516期间,指令1516还可以完全地或部分地驻留在存储器1532内、存储单元1536内、处理器1510中的至少一个内(例如,处理器的高速缓存存储器内)、或其任何合适的组合内。因此,存储器1532、存储单元1536和处理器1510的存储器是机器可读介质的示例。

[0200] 如本文所使用,“机器可读介质”是指能够暂时或永久地存储指令和数据的设备,并且可以包括但不限于随机存取存储器(RAM)、只读存储器(ROM)、缓冲存储器、闪存存储器、光学介质、磁性介质、高速缓冲存储器、其它类型的存储器(例如,可擦除可编程只读存

储器 (EEPROM) 或其任何合适的组合。术语“机器可读介质”应被视为包括能够存储指令 1516 的单个介质或多个介质 (例如集中式或分布式数据库、或相关联的缓存和服务)。术语“机器可读介质”还将被视为包括能够存储由机器 (例如, 机器 1500) 执行的指令 (例如, 指令 1516) 的任何介质或多个介质的组合, 使得指令在由机器 1500 的一个或多个处理器 (例如, 处理器 1510) 执行时, 使机器 1500 执行本文所描述的方法中的任何一个或多个。因此, “机器可读介质”指单个存储装置或设备、以及包括多个存储装置或设备的“基于云”的存储系统或存储网络。

[0201] I/O 组件 1550 可以包括用于接收输入、提供输出、生成输出、发送信息、交换信息、捕捉测量等的各种各样的组件。包括在特定机器中的特定 I/O 组件 1550 将取决于机器的类型。例如, 诸如移动电话的便携式机器将可能包括触摸输入设备或其他这样的输入机构, 而无头服务器机器将可能不包括这样的触摸输入设备。应当理解, I/O 组件 1550 可以包括图 15 中未示出的许多其他组件。I/O 组件 1550 根据功能被分组, 以便简化以下讨论, 并且分组不以任何方式进行限制。在各种示例实施例中, I/O 组件 1550 可以包括输出组件 1552 和输入组件 1554。输出组件 1552 可以包括视觉组件 (例如, 显示器, 诸如等离子体显示面板 (PDP)、发光二极管 (LED) 显示器、液晶显示器 (LCD)、投影仪或阴极射线管 (CRT))、声学组件 (例如扬声器)、触觉组件 (例如振动马达、电阻机构)、其他信号发生器等。输入组件 1554 可以包括字母数字输入组件 (例如, 键盘、配置为接收字母数字输入的触摸屏、光电键盘或其他字母数字输入组件)、基于点的输入组件 (例如, 鼠标、触摸板、轨迹球、操纵杆、运动传感器或其他定点仪器)、触觉输入组件 (例如, 物理按钮、提供触摸或触摸手势的位置和/或力的触摸屏或其他触觉输入组件)、音频输入组件 (例如, 麦克风) 等。

[0202] 在另一些示例实施例中, I/O 组件 1550 可以包括生物测定组件 1556、运动组件 15515、环境组件 1560 或定位组件 1562、以及许多其他组件。例如, 生物统计组件 1556 可包括用于检测表现 (例如, 手表现、面部表现、语音表现、身体姿势或眼睛跟踪)、测量生物信号 (例如, 血压、心率、体温、汗水或脑波)、标识人 (例如, 语音标识、视网膜标识、面部标识、指纹标识或基于脑电图的标识) 等的组件。运动组件 15515 可包括加速度传感器组件 (例如, 加速度计)、重力传感器组件、旋转传感器组件 (例如, 陀螺仪) 等。环境组件 1560 可以包括例如照明传感器组件 (例如光度计)、温度传感器组件 (例如, 检测环境温度的一个或多个温度计)、湿度传感器组件、压力传感器组件 (例如气压计)、声学传感器组件 (例如, 检测背景噪声的一个或多个麦克风)、接近传感器组件 (例如, 检测附近物体的红外传感器)、气体传感器 (例如, 为了安全而检测危险气体的浓度或测量大气中污染物的气体检测传感器) 或可以提供与周围物理环境相对应的指示、测量或信号的其他组件。定位组件 1562 可以包括位置传感器组件 (例如, GPS 接收器组件)、高度传感器组件 (例如, 高度计或气压计, 其检测可以从其导出高度的气压)、取向传感器组件 (例如, 磁力计) 等。

[0203] 可以使用各种各样的技术来实现通信。I/O 组件 1550 可以包括通信组件 1564, 通信组件 1564 可操作以分别经由耦合 1582 和耦合 1572 将机器 1500 耦合到网络 15150 或设备 1570。例如, 通信组件 1564 可以包括网络接口组件或与网络 104 接口连接的其他合适设备。在另一些示例中, 通信组件 1564 可包括有线通信组件、无线通信组件、蜂窝通信组件、近场通信 (NFC) 组件、蓝牙® 组件 (例如 蓝牙® 低能)、Wi-Fi® 组件、以及经由其他模态提供通信的其他通信组件。设备 1570 可以是另一个机器或者各种外围设备中的任何一种 (例如, 通过

USB耦合的外围设备)。

[0204] 此外,通信组件1564可以检测标识符或包括可操作以检测标识符的组件。例如,通信组件1564可以包括射频识别(RFID)标签读取器组件、NFC智能标签检测组件、光学读取器组件(例如,用于检测以下各项的光学传感器:一维条形码(例如通用产品代码(UPC)条形码)、多维条形码(例如快速响应(QR)码)、阿兹台克码、数据矩阵、Dataglyph、MaxiCode、PDF417、超码、UCC RSS-2D条形码和其他光学码)、或声学检测组件(例如,用于识别已标记的音频信号的麦克风)。此外,可以经由通信组件1564来导出各种信息,例如经由互联网协议(IP)地理位置的位置、经由Wi-Fi®信号三角测量的位置、经由检测可以指示特定位置的NFC信标信号的位置等等。

[0205] 传输介质

[0206] 在各种示例实施例中,网络104的一个或多个部分可以是自组织网络、内联网、外联网、VPN、LAN、WLAN、WAN、WWAN、MAN、互联网,互联网的一部分、PSTN的一部分、普通老式电话服务(POTS)网络、蜂窝电话网络、无线网络、Wi-Fi®网络、另一类型的网络、或两个或更多个这样的网络的组合。例如,网络104或网络104的一部分可以包括无线或蜂窝网络,并且耦合1582可以是码分多址(CDMA)连接、全球移动通信系统(GSM)连接或其他类型的蜂窝或无线耦合。在该示例中,耦合1582可以实现各种类型的数据传输技术中的任何一种,例如单载波无线电传输技术(1xRTT)、演进数据优化(EVDO)技术、通用分组无线电服务(GPRS)技术、GSM演进增强数据速率(EDGE)技术、包括3G的第三代合作伙伴计划(3GPP)、第四代无线(4G)网络、通用移动通信系统(UMTS)、高速分组接入(HSPA)、全球微波接入互操作性(WiMAX)、长期演进(LTE)标准、由各种标准设置组织定义的其他标准、其他远程协议或其他数据传输技术。

[0207] 可以经由网络接口设备(例如,包括在通信组件1564中的网络接口组件)使用传输介质并且利用多个公知的传输协议(例如,超文本传输协议(HTTP))通过网络104发送或接收指令1516。类似地,可以使用传输介质经由耦合1572(例如,对等耦合)向设备1570发送或从其接收指令1516。术语“传输介质”应被认为包括能够存储、编码或承载用于被机器1500执行的指令1516的任意无形介质,并且包括用于促进该软件的通信的数字或模拟通信信号或其他无形介质。传输介质是机器可读介质的一个实施例。

[0208] 示例方法

[0209] 图16示出了识别发布的相关类别的示例方法1600。方法1600包括:操作1610,访问对将发布添加到发布语料库的请求;操作1620,识别发布的相关类别集合;以及操作1630,显示发布的相关类别集合。

[0210] 操作1610利用一个或多个处理器访问来自用户设备的请求将发布添加到发布语料库并且识别所述发布的相关类别集合的请求。例如,在图2中,列表系统150的服务器中的一个或多个处理器访问来自用户设备204的请求。图3B是通过来自用户设备的请求添加的发布的示例。

[0211] 操作1620利用所述一个或多个处理器,通过在以下二者之间进行比较来识别一个或多个最接近的匹配:(i)与所述发布的至少一部分相对应的发布语义向量,所述发布语义向量基于将所述发布的所述至少一部分投射到语义向量空间中的第一机器学习模型,以及(ii)与来自多个类别的相应类别相对应的多个类别向量,所述多个类别向量基于将所述多

个类别投射到所述语义向量空间中的第二机器学习模型,所述多个类别是所述发布在所述发布语料库中的分类。图4是识别最接近的匹配的示例。

[0212] 操作1630使得在所述用户设备上显示所述一个或多个最接近的匹配,作为所述发布语料库的所述相关类别集合。例如,在图2中,列表系统50的服务器中的一个或多个处理器使得在用户设备204上进行显示。图3A是最接近匹配的示例显示。

[0213] 语言

[0214] 在整个说明书中,复数实例可以实现如单个实例所描述的部件、操作或结构。虽然一个或多个方法的各个操作被示意和描述为分离的操作,但是各个操作中的一个或多个可以同时执行,并且无需按所示顺序执行操作。在示例配置中被示为分离组件的结构和功能可以被实现为组合结构或组件。类似地,被示为单个组件的结构和功能可以被实现为分离的组件。这些和其他变型、修改、添加和改进落入本文中主题的范围之内。

[0215] 尽管已经参考具体示例实施例描述了本发明主题的概述,但是在不脱离本公开的实施例的更宽范围的情况下,可以对这些实施例进行各种修改和改变。本发明主题的这些实施例在本文中可以单独地或共同地由术语“发明”提及,以仅仅为了方便,并且不旨在自动地将本申请的范围限制为任何单个公开或发明构思(如果事实上公开了一个以上)。

[0216] 充分详细地描述了本文示出的实施例以使本领域技术人员能够实现所公开的教导。可以利用其他实施例并根据这些实施例导出其他实施例,从而可以在不脱离本公开的范围的情况下做出结构和逻辑上的替换和改变。因此,该“具体实施方式”不应当看做是限制意义,并且各种实施例的范围仅通过所附权利要求以及权利要求的等同物的全部范围来限定。

[0217] 如本文所使用的,术语“或”可以被解释为包括性或排他性的意义。此外,可以针对本文中描述为单个实例的资源、操作或结构提供多个实例。另外,各种资源、操作、模块、引擎和数据存储之间的边界在某种程度上是任何的,并且在具体说明性配置的上下文中示出了特定操作。设想了功能的其他分配,并且这些分配可以落入本公开的各种实施例的范围之内。一般来说,在示例配置中作为分离资源呈现的结构和功能可以被实现为组合的结构或资源。类似地,作为单个资源呈现的结构和功能可以被实现为分离的资源。这些和其他变型、修改、添加和改进落入由所附权利要求表示的本公开的实施例的范围之内。因此,说明书和附图应当被看做说明性的而不是限制意义的。

[0218] 下面的编号示例是实施例。

[0219] 1. 一种方法,包括:

[0220] 利用一个或多个处理器访问来自用户设备的请求将发布添加到发布语料库并且识别所述发布的相关类别集合的请求;

[0221] 利用一个或多个处理器,识别以下二者之间的一个或多个最接近的匹配:(i) 与所述发布的至少一部分相对应的发布语义向量,所述发布语义向量基于将所述发布的所述至少一部分投射到语义向量空间中的第一机器学习模型,以及(ii) 与来自多个类别的相应类别相对应的多个类别向量,所述多个类别向量基于将所述多个类别投射到所述语义向量空间中的第二机器学习模型,所述多个类别是所述发布在所述发布语料库中的分类;以及

[0222] 使得在所述用户设备上显示所述一个或多个最接近的匹配,作为所述发布语料库的所述相关类别集合。

- [0223] 2. 根据示例1所述的方法,其中,所述类别是叶类别。
- [0224] 3. 根据示例1或示例2所述的方法,其中,所述类别是在所述多个类别的类别树中的根级之下的至少两个树级的类别路径。
- [0225] 4. 根据示例1至3中任一项所述的方法,其中,所述发布的所述至少一部分包括所述发布的标题。
- [0226] 5. 根据示例 1所述的方法,其中,在从所述发布语料库的先前添加的发布中自动导出的数据上训练所述第一机器学习模型和所述第二机器学习模型中的至少一个。
- [0227] 6. 根据示例1至5中任一项所述的方法,其中,在子词级和字符级别中的一个或多个处训练所述第一机器学习模型和所述第二机器学习模型中的至少一个,以减少运行时的词汇外术语。
- [0228] 7. 根据示例1至6中任一项所述的方法,还包括:
- [0229] 向所述多个类别添加新类别,而不在所述新类别上重新训练所述第二机器学习模型,
- [0230] 其中,被识别为一个或多个最接近的匹配的所述一个或多个最接近的匹配包括所述新类别。
- [0231] 8. 一种计算机,包括:
- [0232] 存储指令的存储设备;以及
- [0233] 一个或多个硬件处理器,由所述指令配置为执行包括以下各项的操作:
- [0234] 利用一个或多个处理器访问来自用户设备的请求将发布添加到发布语料库并且识别所述发布的相关类别集合的请求;
- [0235] 利用一个或多个处理器,识别以下二者之间的一个或多个最接近的匹配:(i) 与所述发布的至少一部分相对应的发布语义向量,所述发布语义向量基于将所述发布的所述至少一部分投射到语义向量空间中的第一机器学习模型,以及(ii) 与来自多个类别的相应类别相对应的多个类别向量,所述多个类别向量基于将所述多个类别投射到所述语义向量空间中的第二机器学习模型,所述多个类别是所述发布在所述发布语料库中的分类;以及
- [0236] 使得在所述用户设备上显示所述一个或多个最接近的匹配,作为所述发布语料库的所述相关类别集合。
- [0237] 9. 根据示例8所述的计算机,其中,所述类别是叶类别。
- [0238] 10. 根据示例8或示例9所述的计算机,其中,所述类别是在所述多个类别的类别树中的根级之下的至少两个树级的类别路径。
- [0239] 11. 根据示例8至10中任一项所述的计算机,其中,所述发布的所述至少一部分包括所述发布的标题。
- [0240] 12. 根据示例8至11中任一项所述的计算机,其中,在从所述发布语料库的先前添加的发布中自动导出的数据上训练所述第一机器学习模型和所述第二机器学习模型中的至少一个。
- [0241] 13. 根据示例8至12中任一项所述的计算机,其中,在子词级和字符级别中的一个或多个处训练所述第一机器学习模型和所述第二机器学习模型中的至少一个,以减少运行时的词汇外术语。
- [0242] 14. 根据示例 8所述的计算机,所述操作还包括:

[0243] 向所述多个类别添加新类别,而不在所述新类别上重新训练所述第二机器学习模型,

[0244] 其中,被识别为一个或多个最接近的匹配的所述一个或多个最接近的匹配包括所述新类别。

[0245] 15.一种存储指令的硬件机器可读设备,所述指令在被机器的一个或多个处理器执行时使得所述机器执行包括以下各项的操作:

[0246] 利用一个或多个处理器访问来自用户设备的请求将发布添加到发布语料库并且识别所述发布的相关类别集合的请求;

[0247] 利用一个或多个处理器,识别以下二者之间的一个或多个最接近的匹配:(i)与所述发布的至少一部分相对应的发布语义向量,所述发布语义向量基于将所述发布的所述至少一部分投射到语义向量空间中的第一机器学习模型,以及(ii)与来自多个类别的相应类别相对应的多个类别向量,所述多个类别向量基于将所述多个类别投射到所述语义向量空间中的第二机器学习模型,所述多个类别是所述发布在所述发布语料库中的分类;以及

[0248] 使得在所述用户设备上显示所述一个或多个最接近的匹配,作为所述发布语料库的所述相关类别集合。

[0249] 16.根据示例15所述的计算机,其中,所述类别是叶类别。

[0250] 17.根据示例15或示例16所述的计算机,其中,所述类别是在所述多个类别的类别树中的根级之下的至少两个树级的类别路径。

[0251] 18.根据示例15至17中任一项所述的计算机,其中,所述发布的所述至少一部分包括所述发布的标题。

[0252] 19.根据示例15至18中任一项所述的计算机,其中,在从所述发布语料库的先前添加的发布中自动导出的数据上训练所述第一机器学习模型和所述第二机器学习模型中的至少一个。

[0253] 20.根据示例15至19中任一项所述的计算机,其中,在子词级和字符级别中的一个或多个处训练所述第一机器学习模型和所述第二机器学习模型中的至少一个,以减少运行时的词汇外术语。

[0254] 21.一种携带机器可读指令的机器可读介质,所述机器可读指令在被机器的一个或多个处理器执行时,使所述机器执行根据示例1至7中任一项所述的方法。

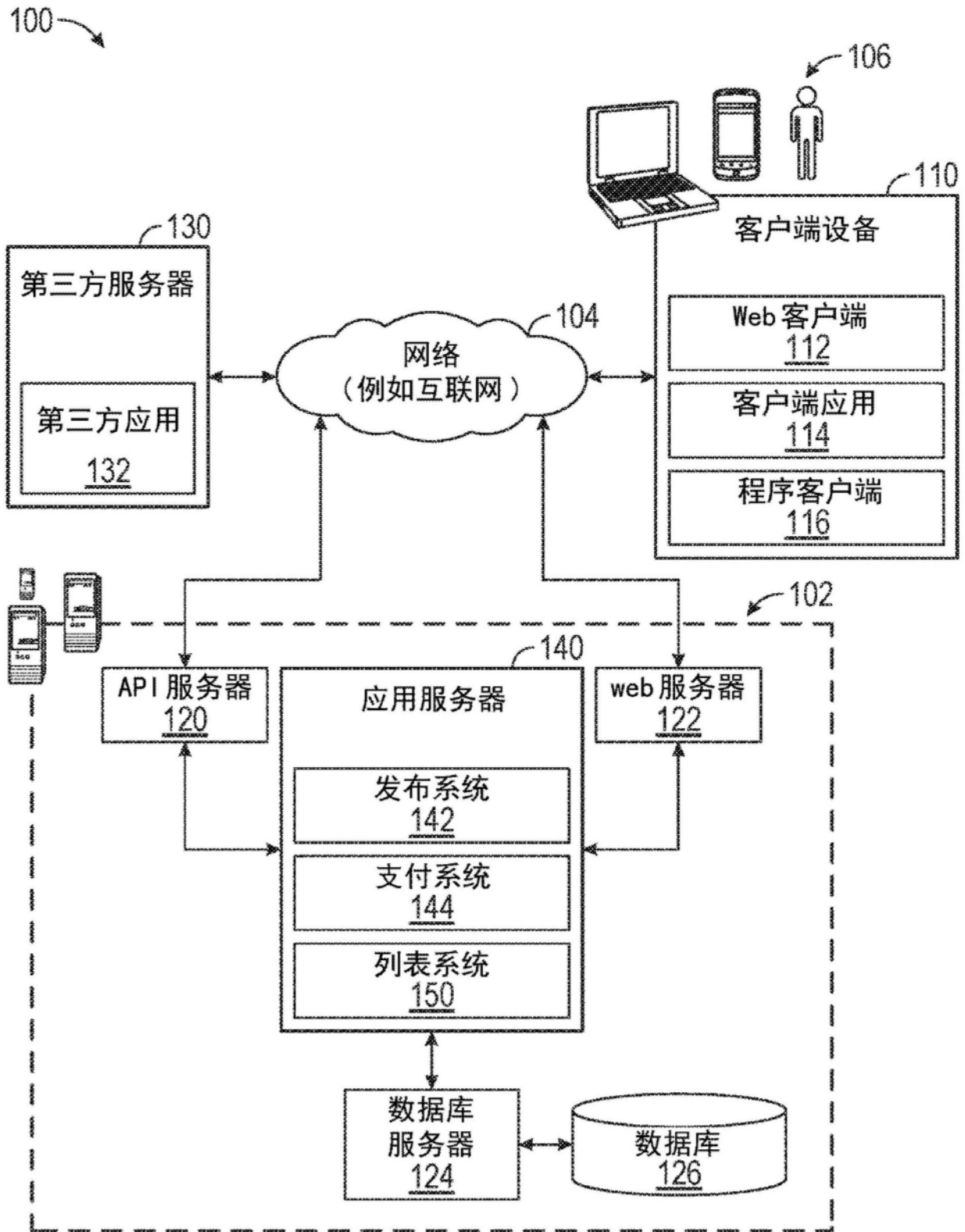


图1

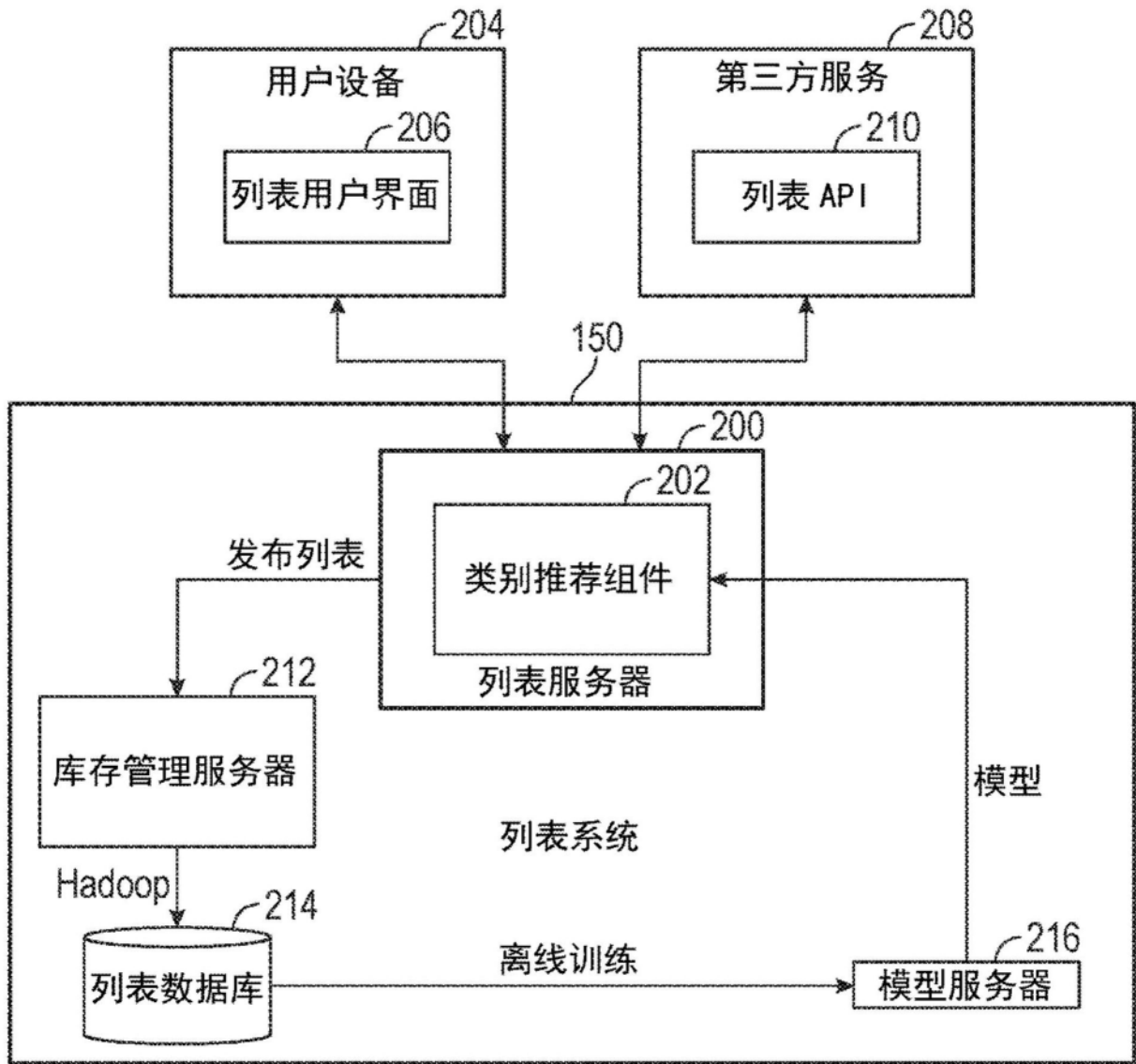


图2

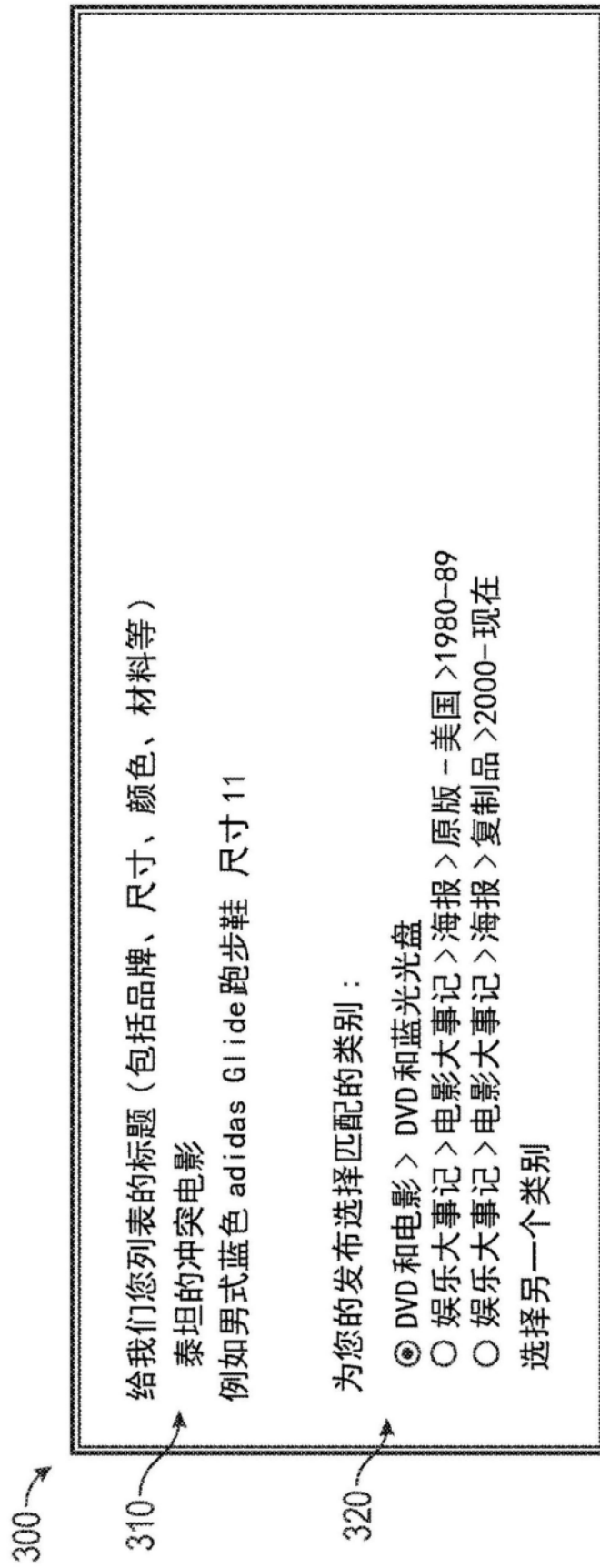


图3A

Hi Kenneth! ▾ | Daily Deals | Gift Cards | Sell | Help & Contact DEAL FRENZY UP TO 70% OFF

Shop by ▾ category Search...

⌂ Back to home page | Listed as Clash of the Titans (Blu-ray Disc, 2010) in category: DVDs & Movies > DVDs & Blu-ray Discs

Clash of the Titans (Blu-ray Disc, 2010)

Items condition: Brand New
Time left:

Starting bid: **US \$0.99** [0 bids] Place bid
Enter US \$0.99 or more

Price: **US \$5.00** Buy it now Add to cart Add to Watch List ▾




图3B

400
示例：
列表标题：“hello kitty T-shirt” 410
源 (X) 的语义向量：[0.1, 2.3, 3.0] 420

向量空间：
目标 (Y1) 的语义向量：[0.1, 2.2, 3.0] 430
目标 (Y2) 的语义向量：[0.5, 2.6, 2.3] 440

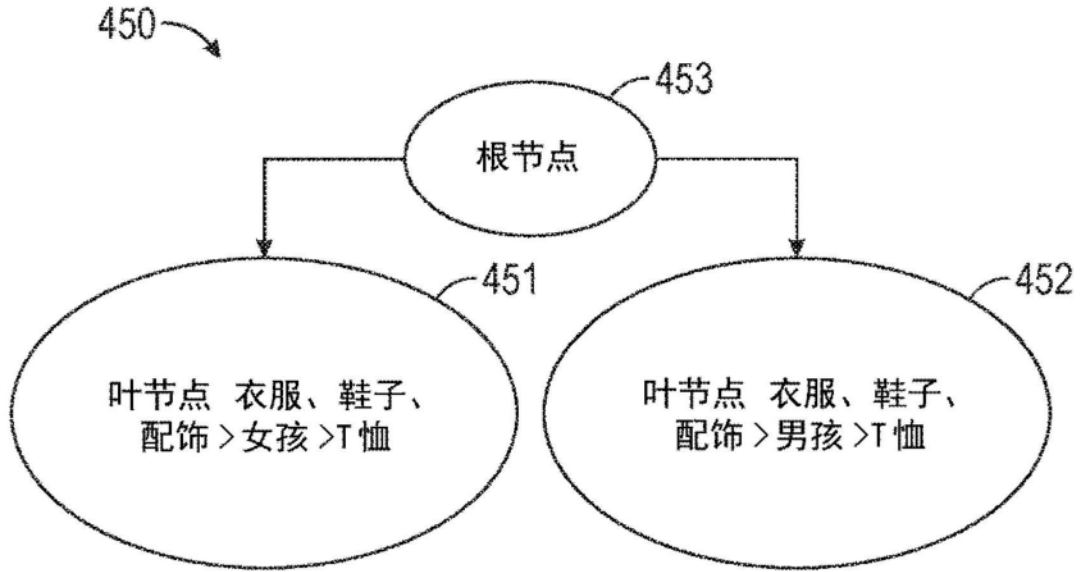


图4

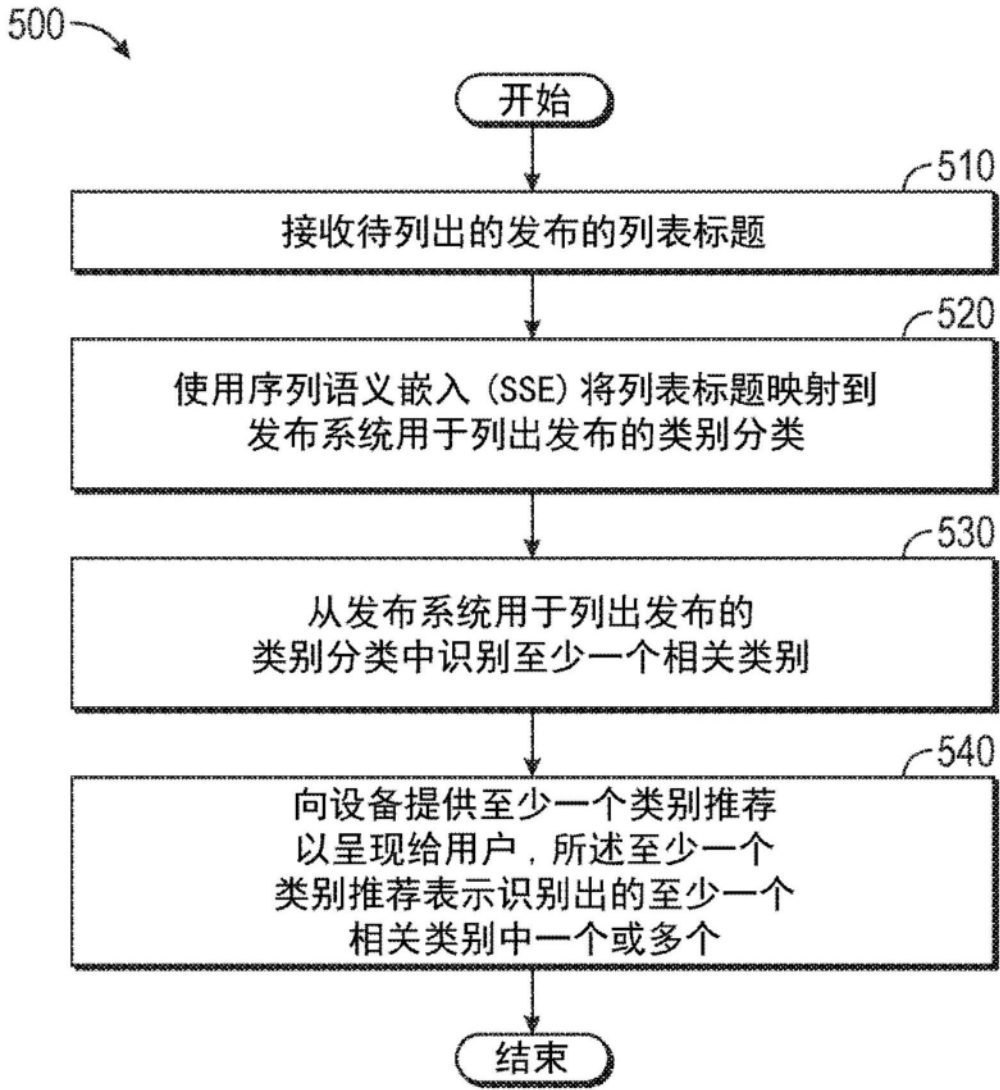


图5A

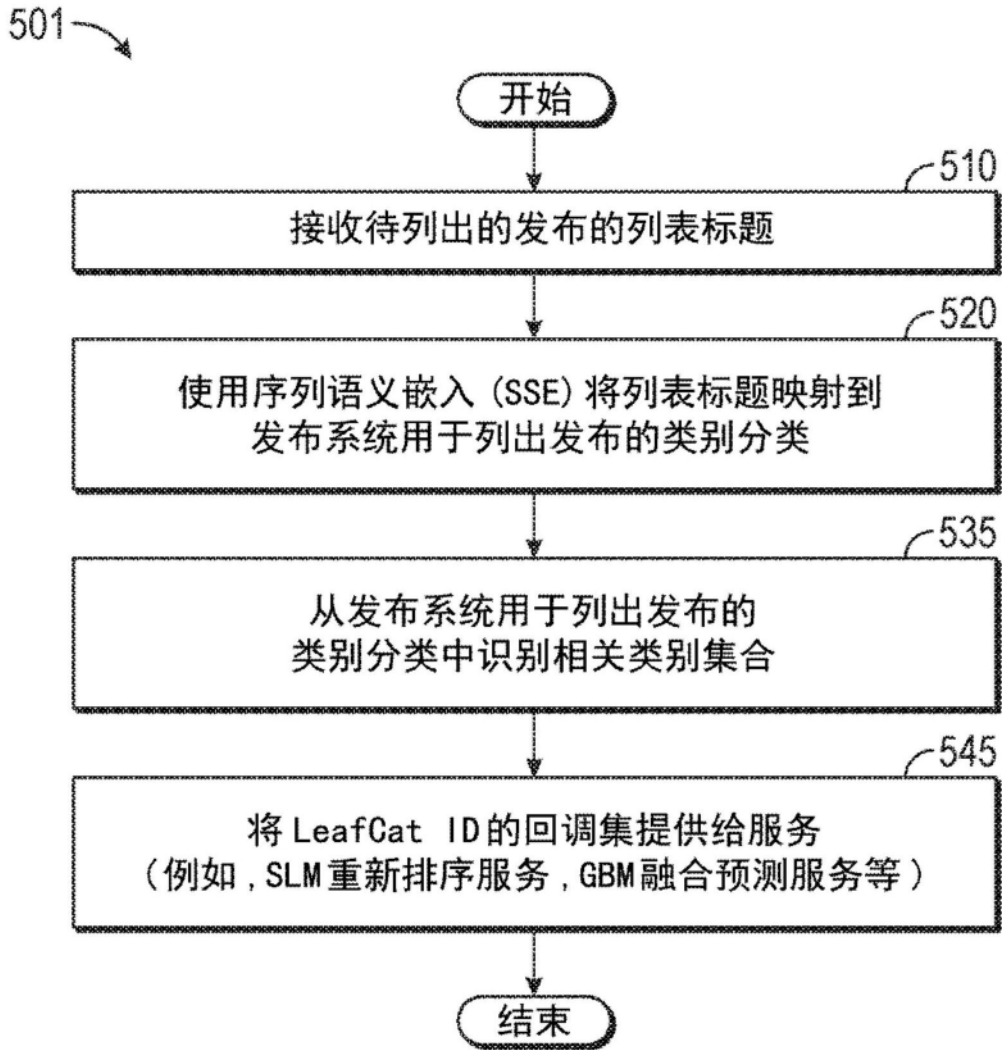


图5B

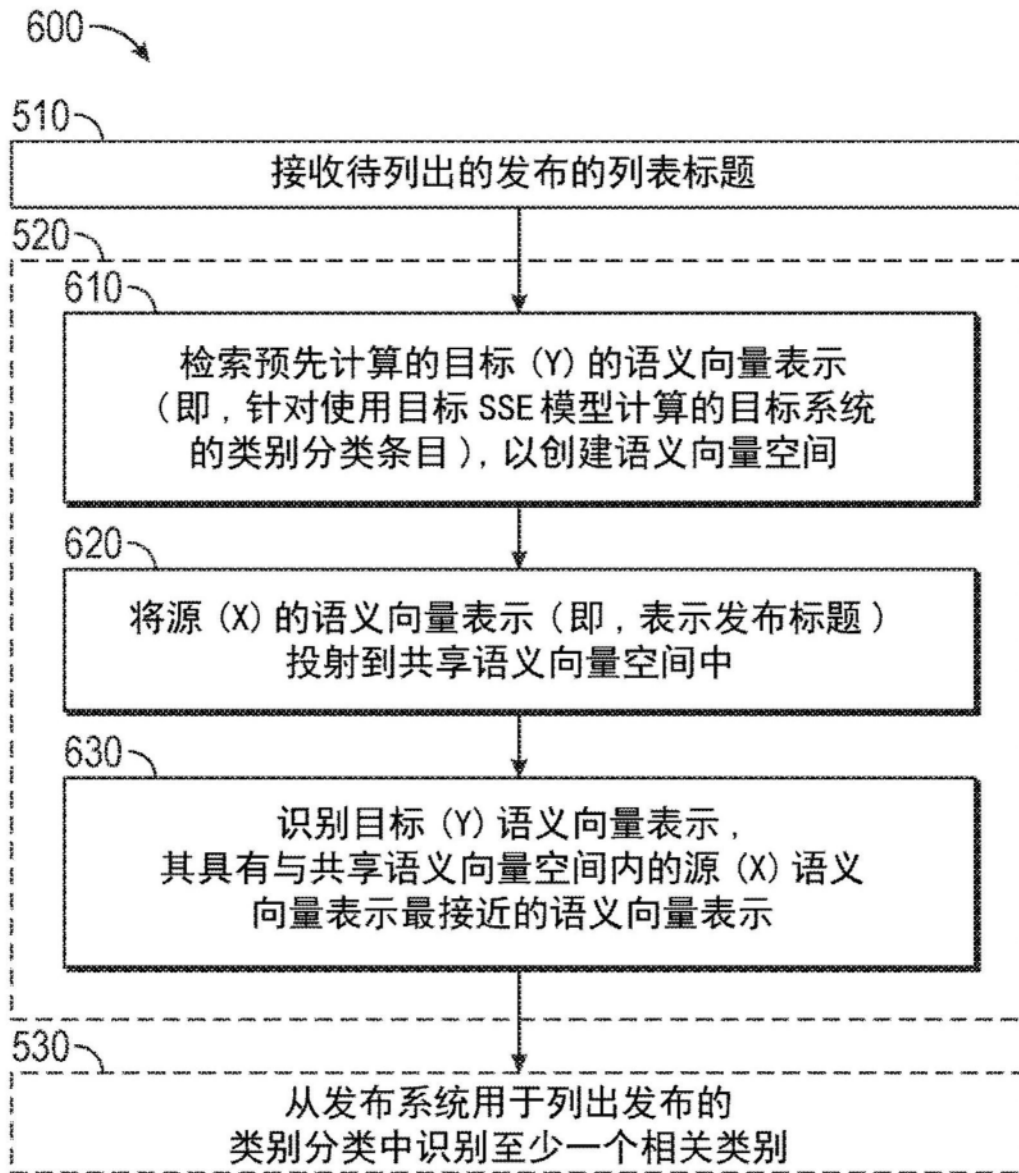


图6A

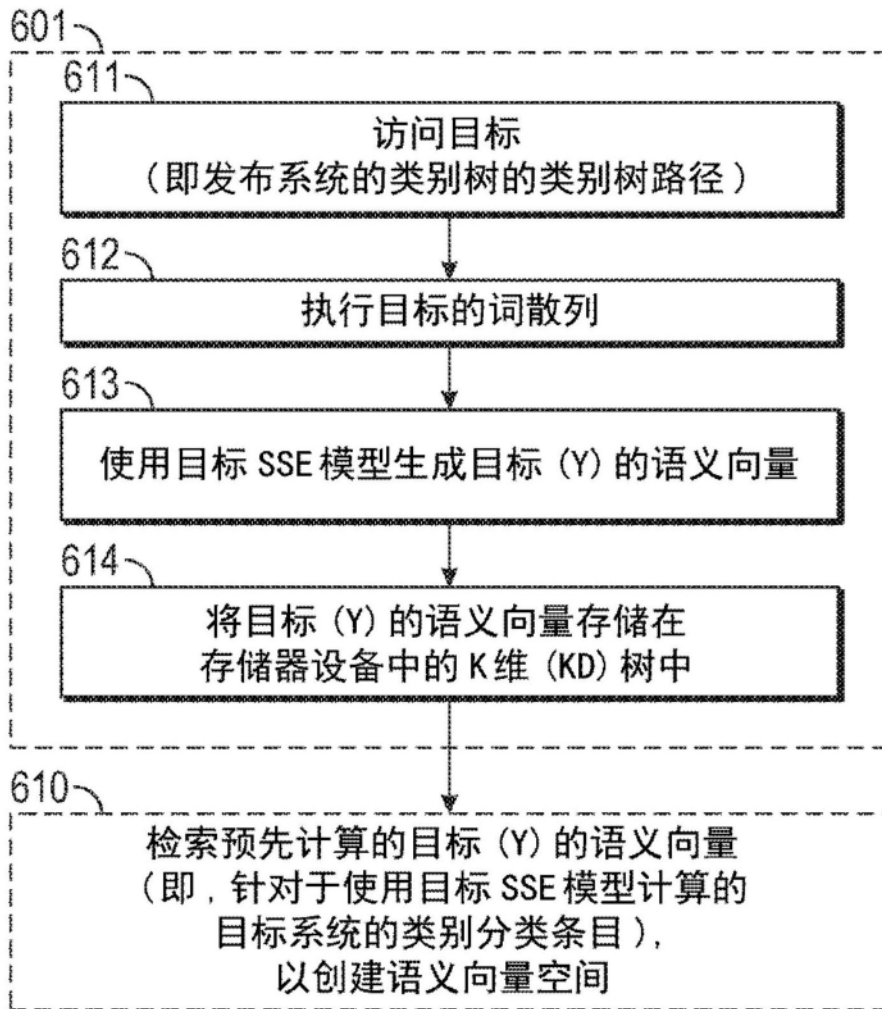


图6B

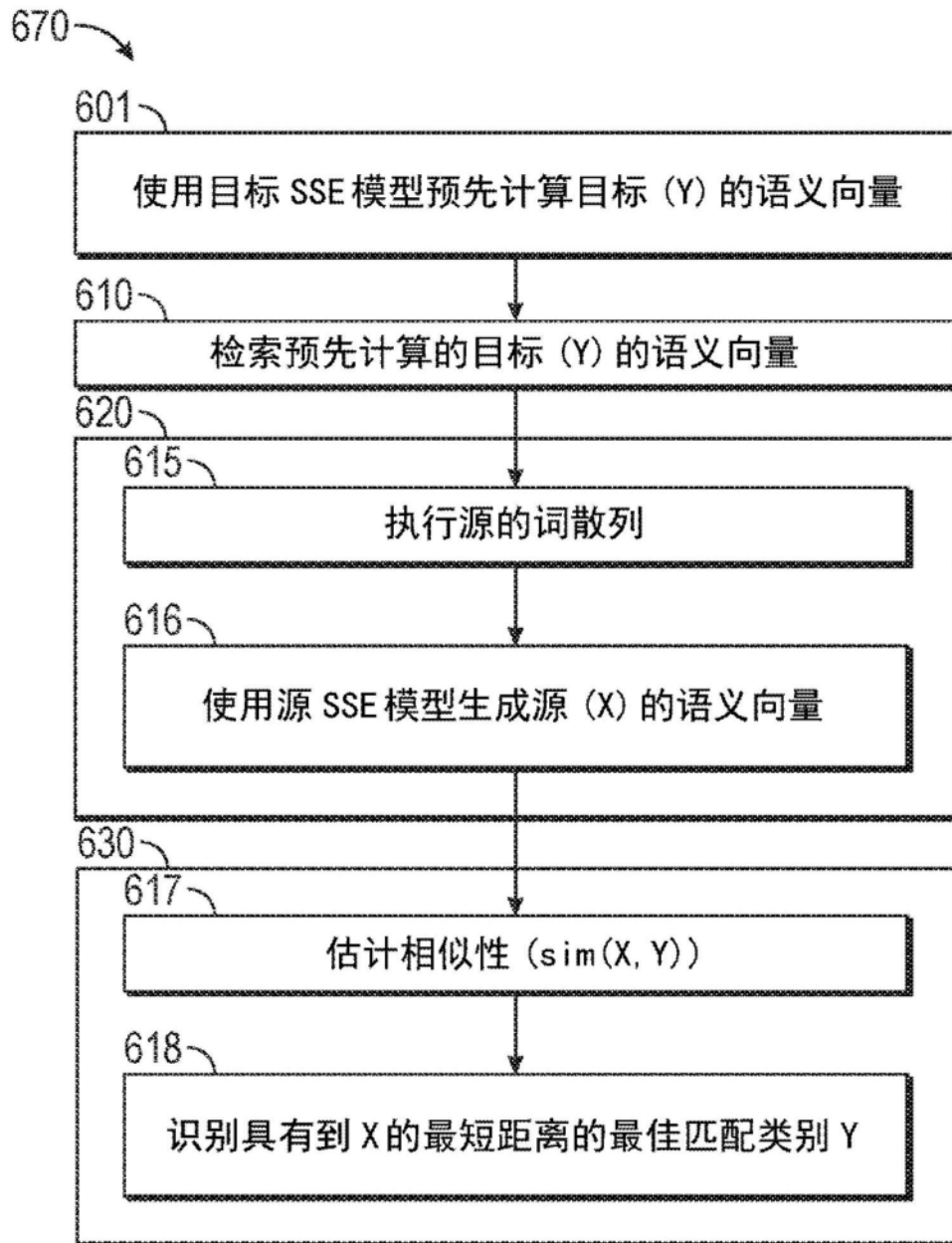


图6C

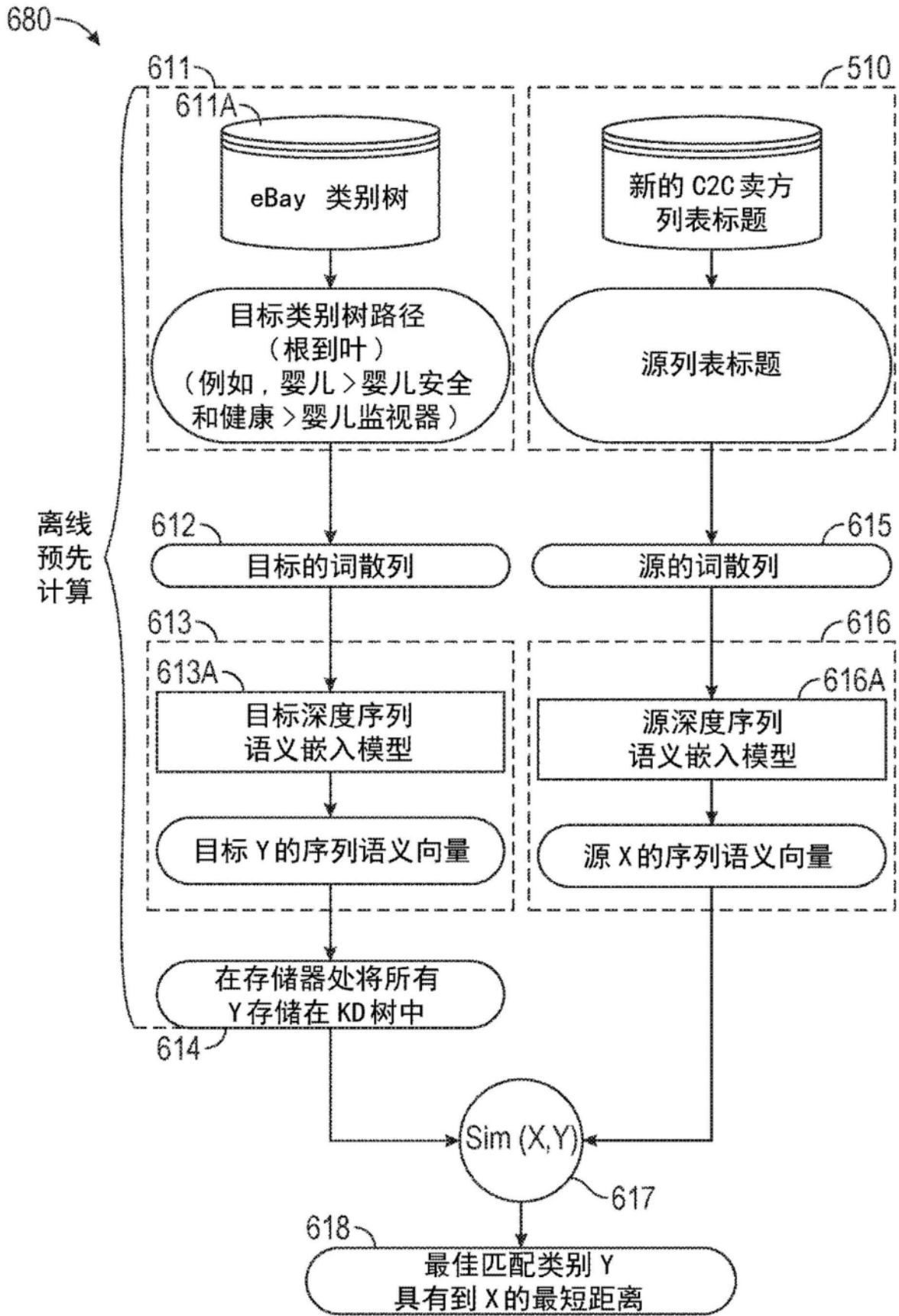


图6D

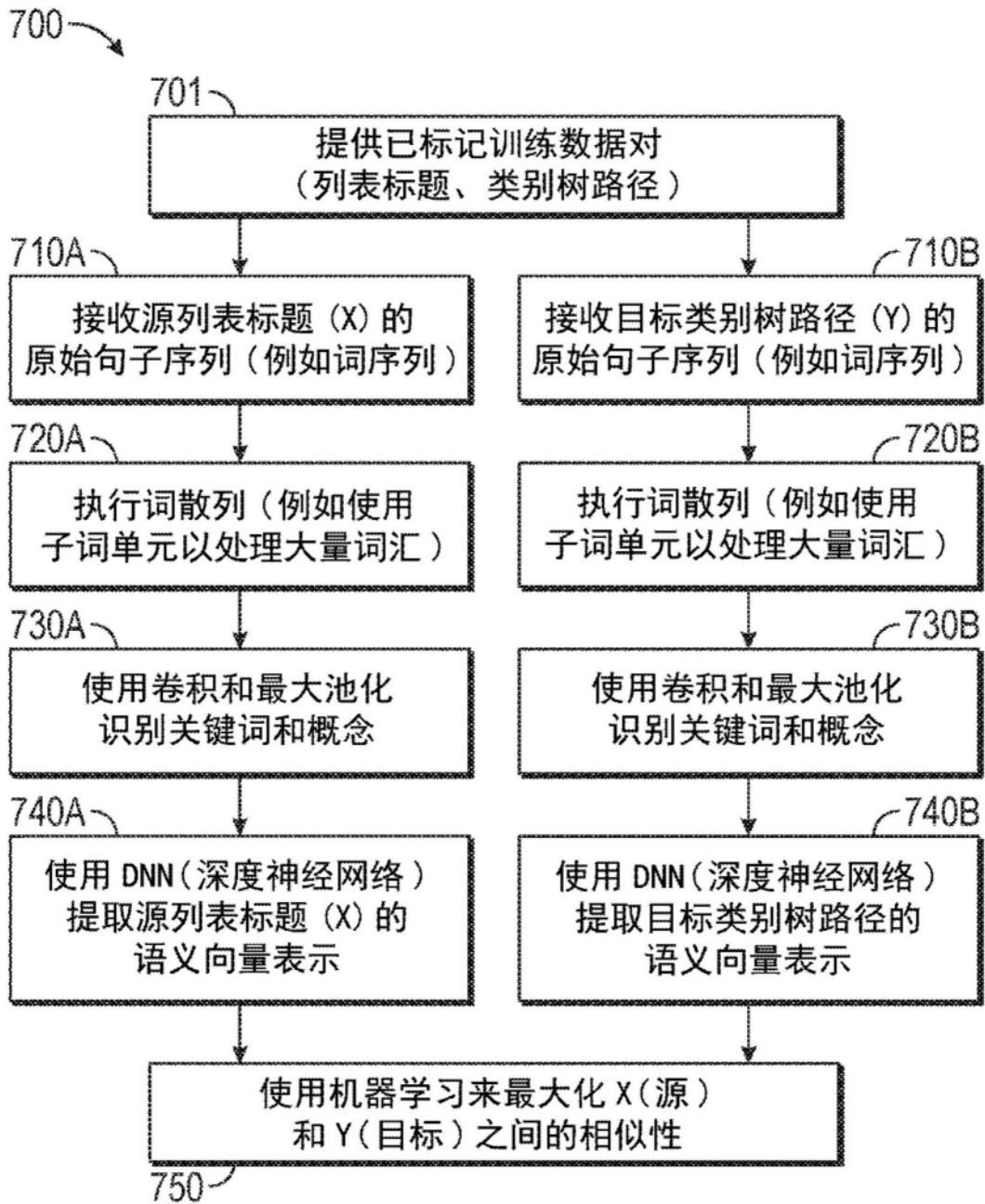


图7

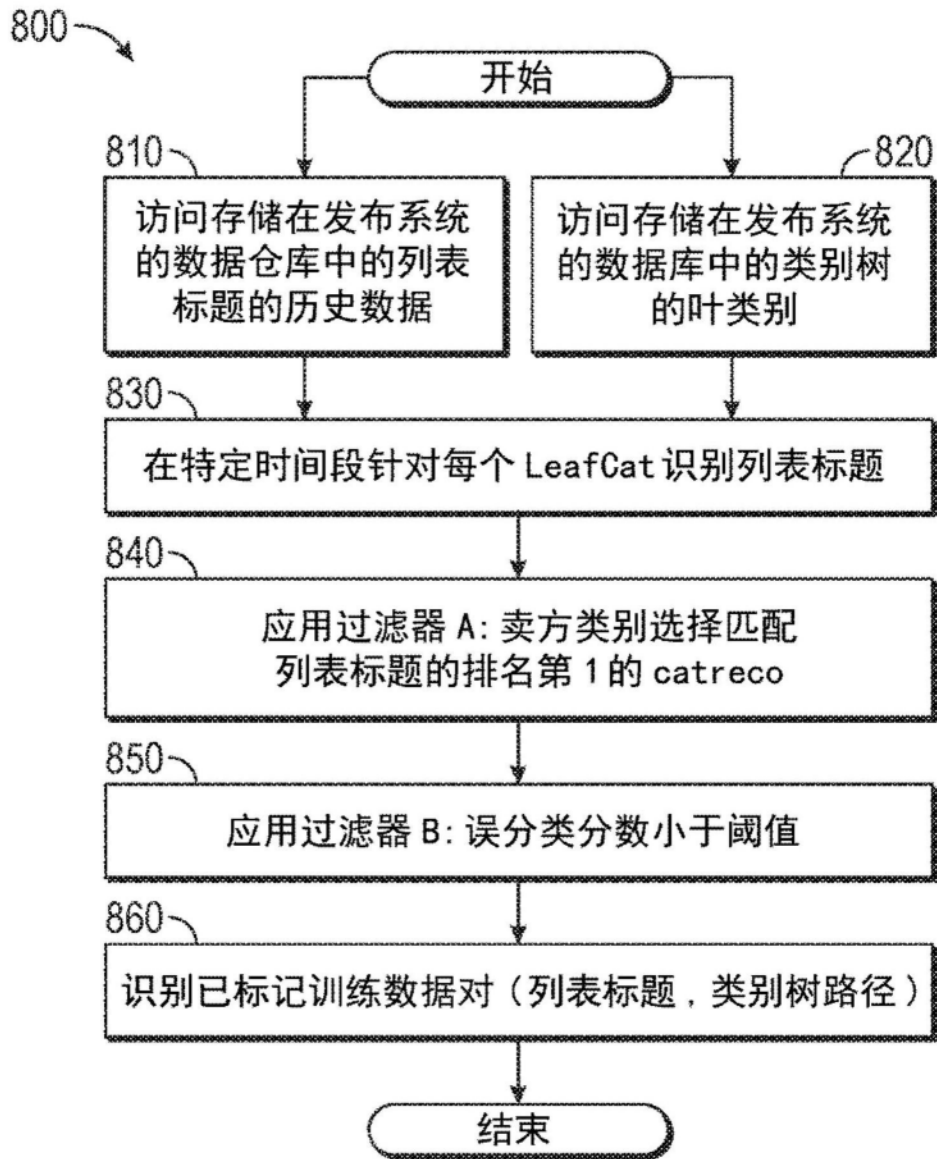


图8

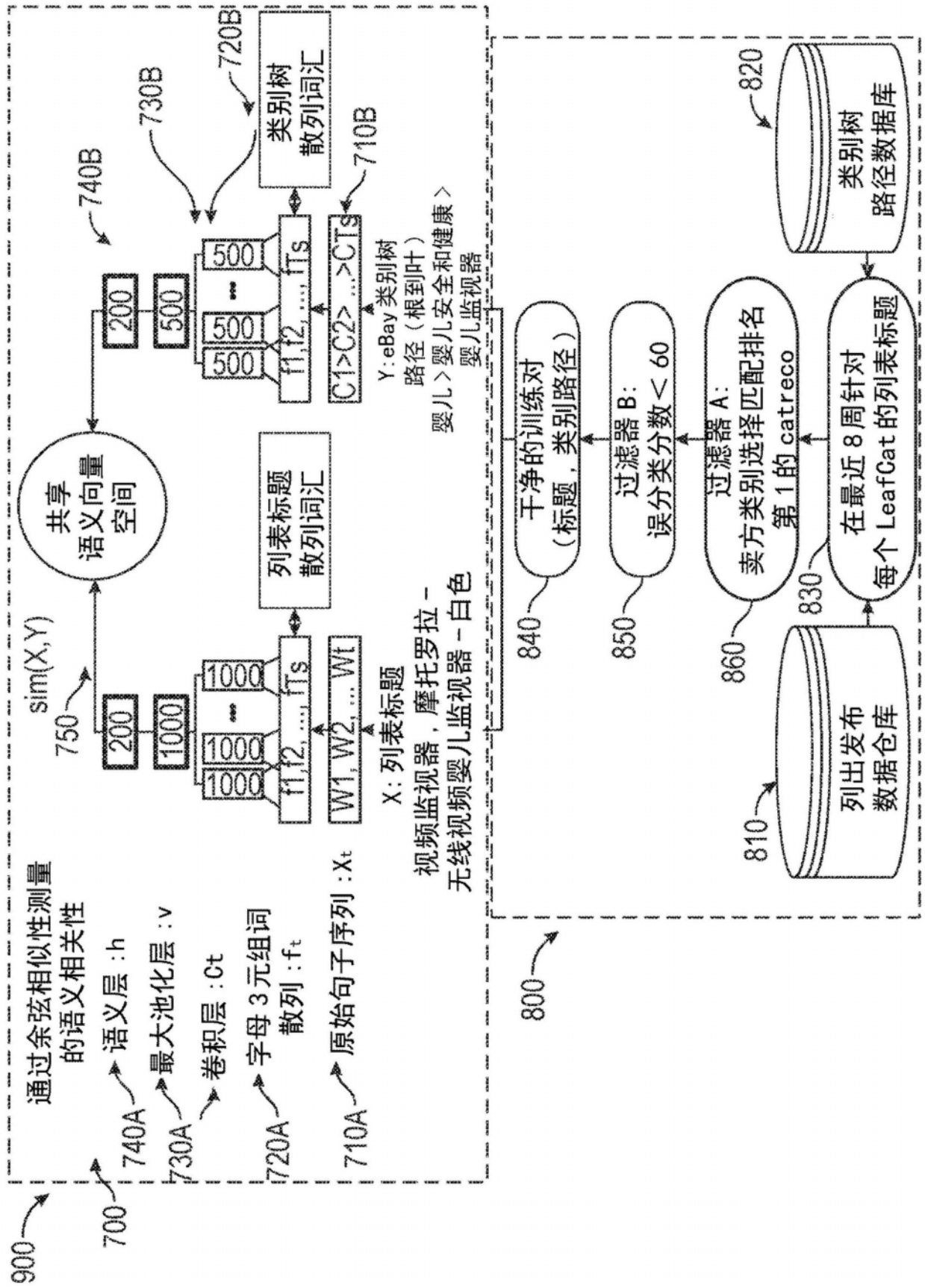


图9

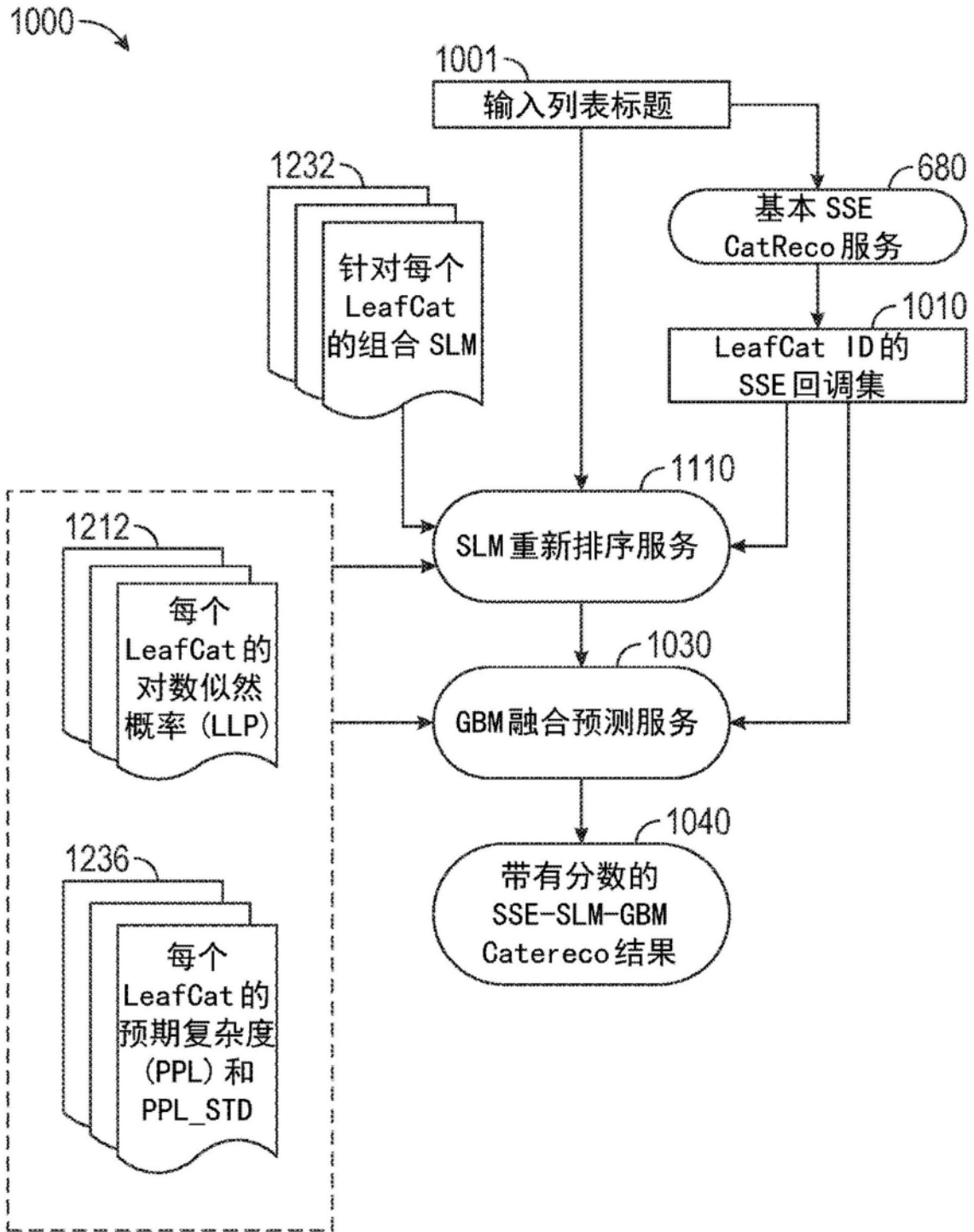


图10

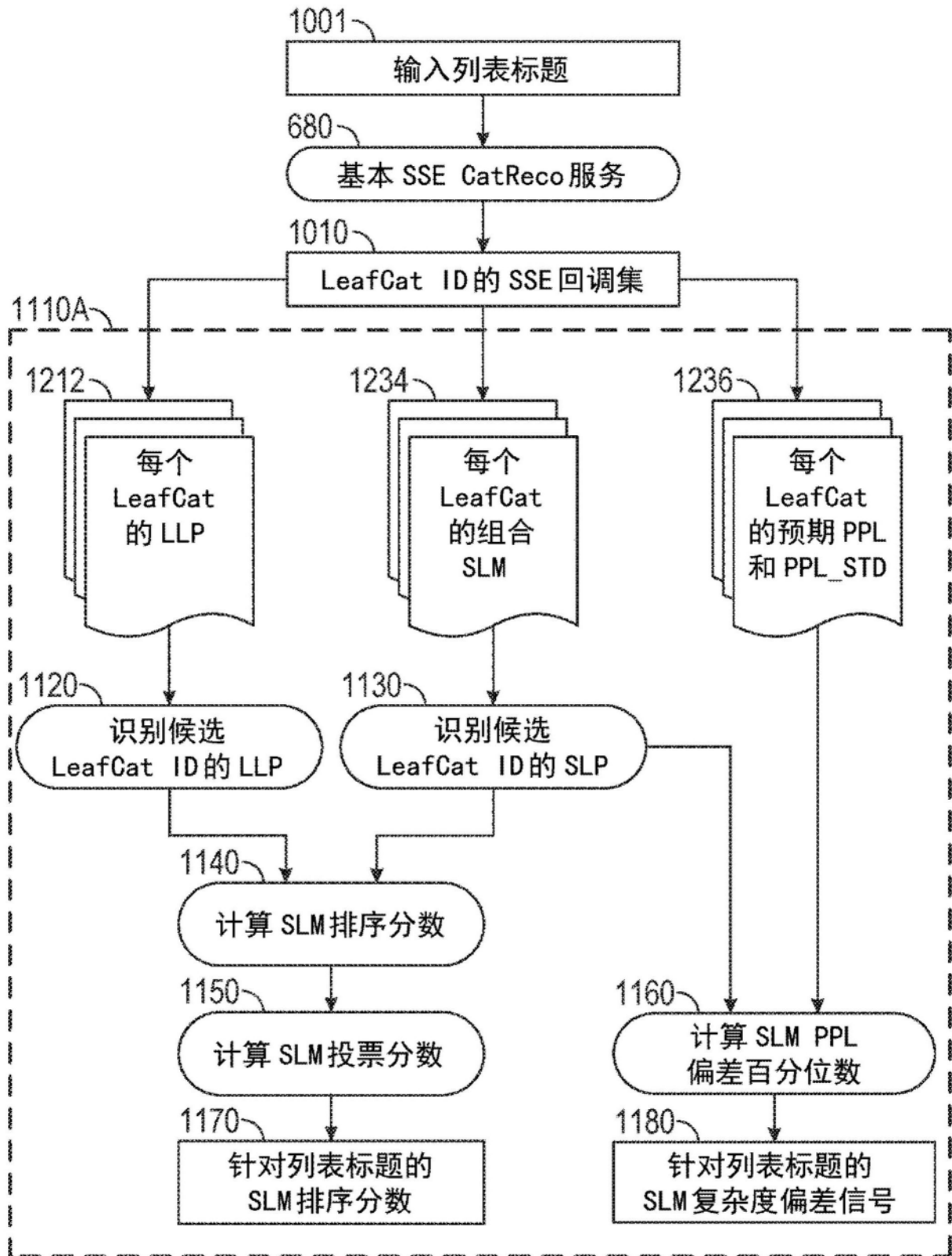


图11

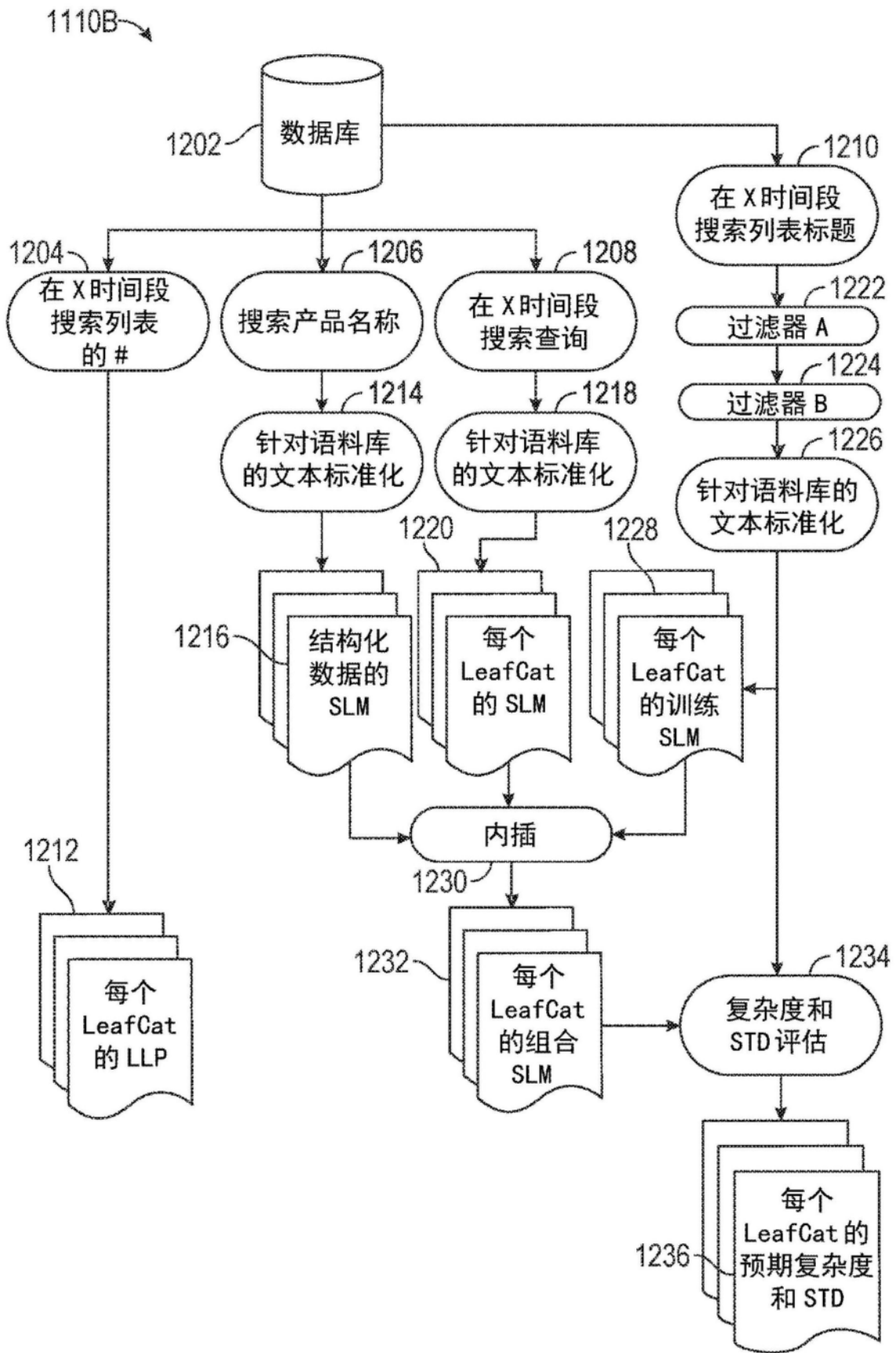


图12

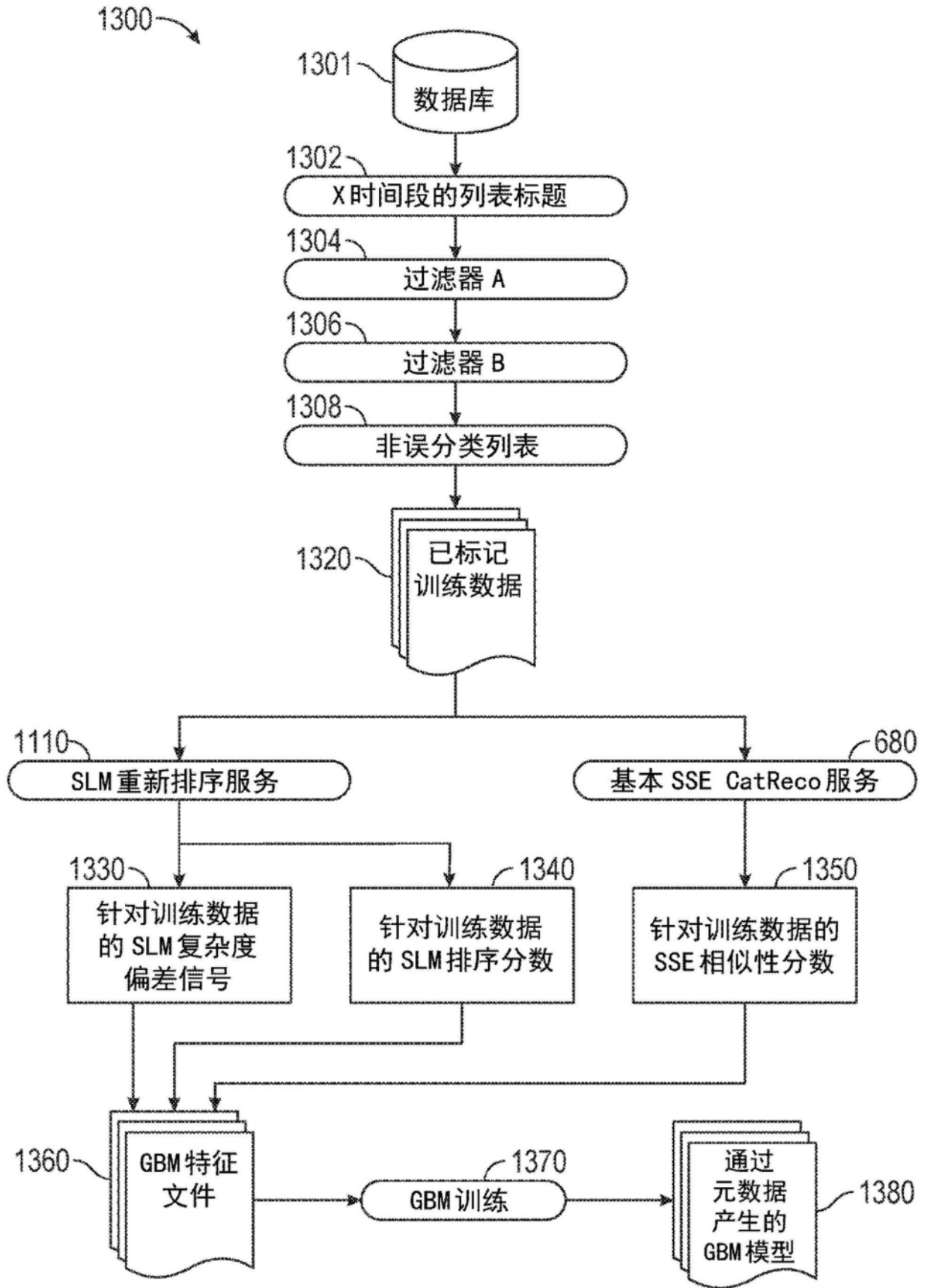


图13

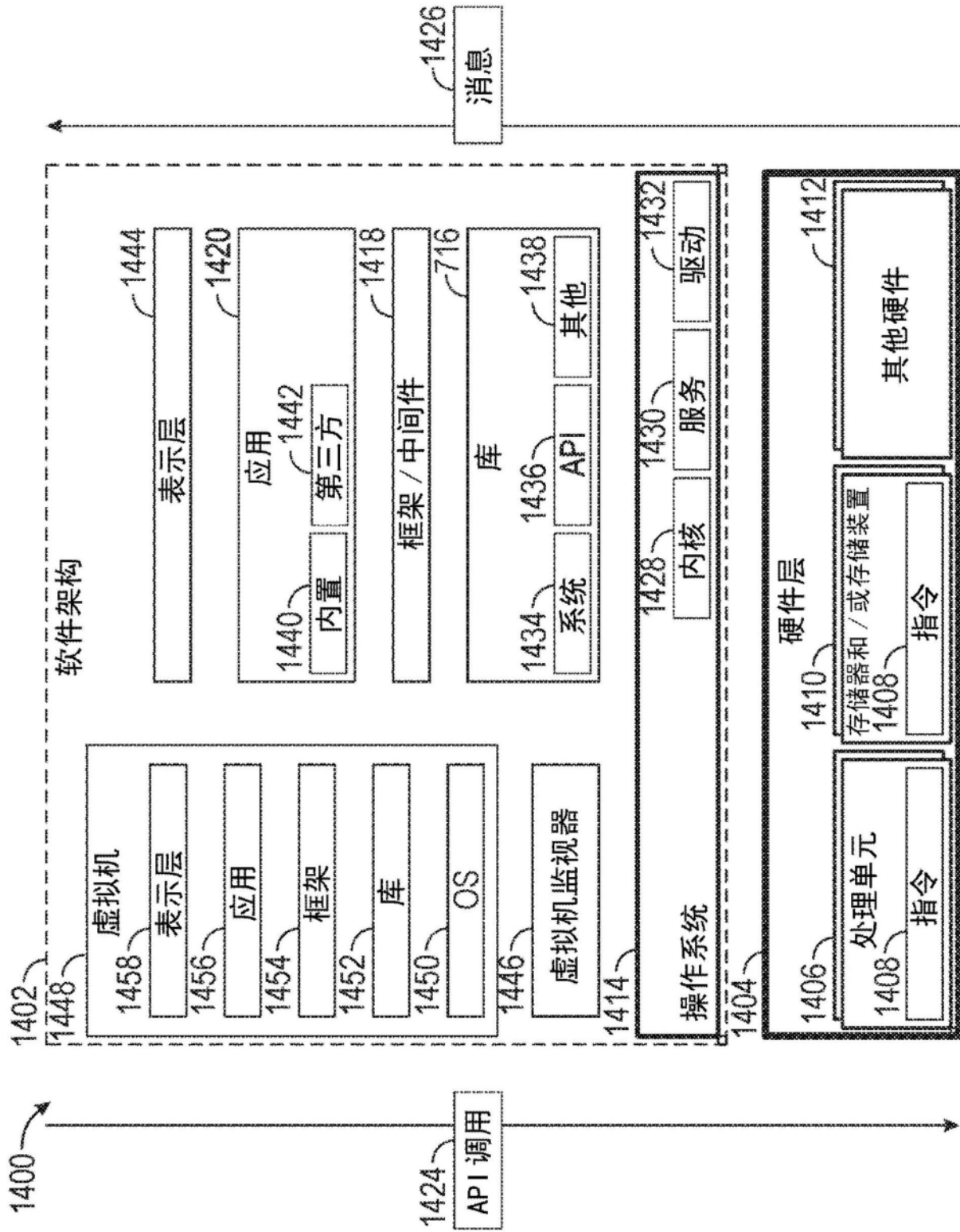


图14

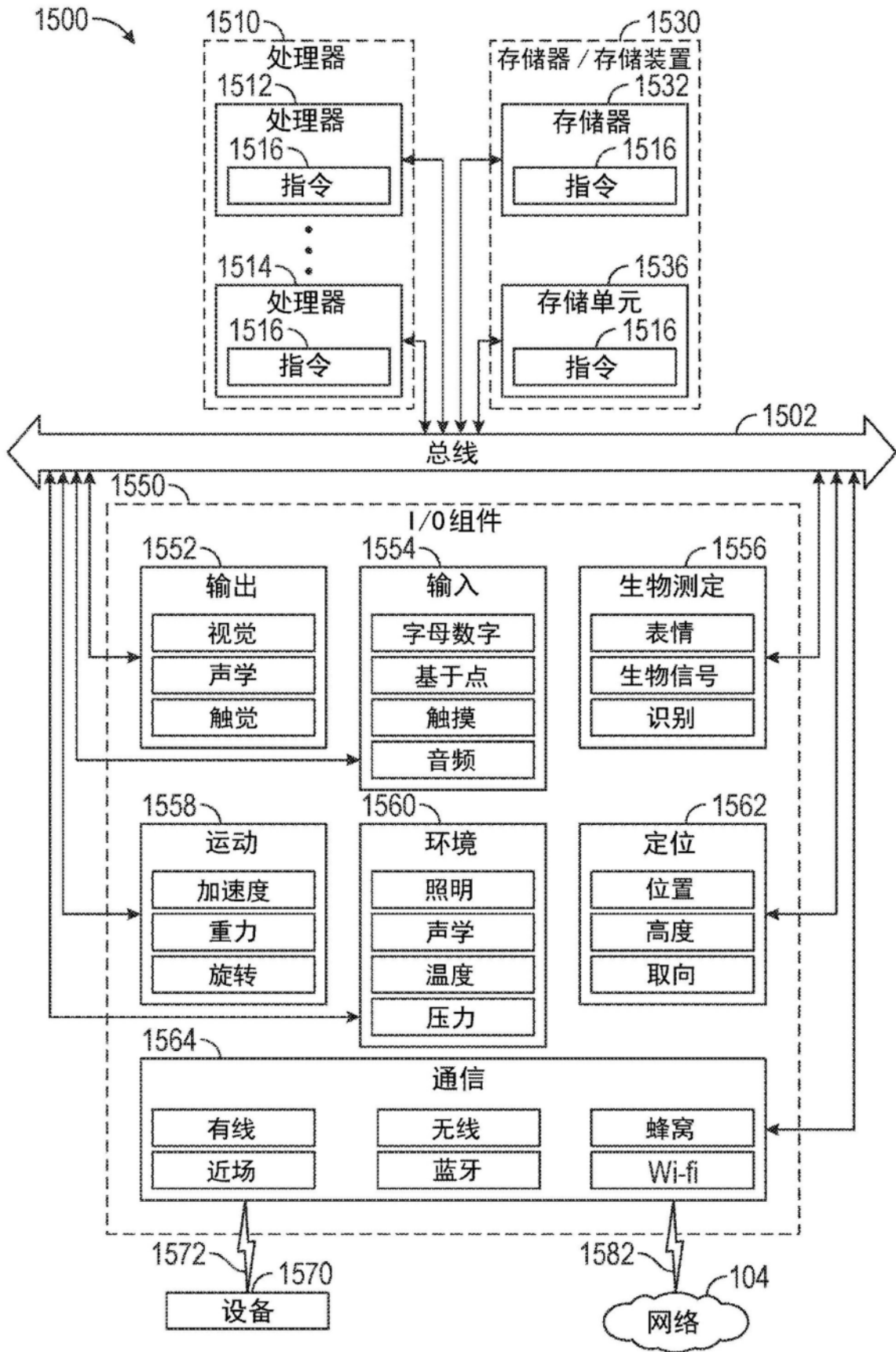


图15

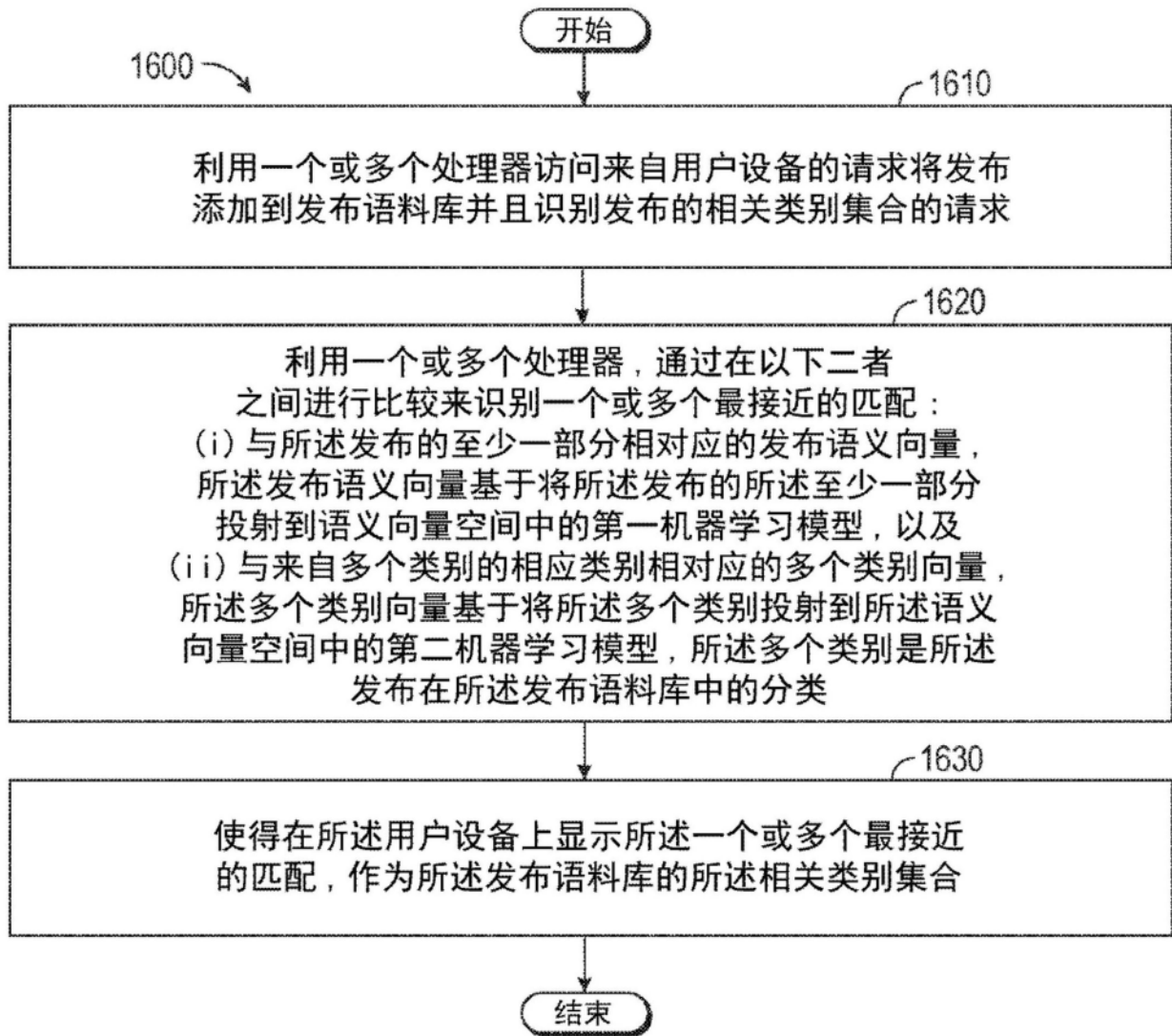


图16